Einführung in MS-DOS

Teil 1

Wolfram Schulze, Uwe Schulze, Berlin

Viele MS-DOS-Einsteiger sind durch die vorhandene Literatur, die zumeist Wert auf Systematik und Vollständigkeit legt, nicht ausreichend bedient. Mit diesem Kurs soll deshalb der Versuch unternommen werden, Ihnen eine didaktische Arbeitshilfe zu bieten, in die die Erfahrungen der Arbeit mit dem Betriebssystem einfließen. Deshalb sind die Befehle auch nicht – wie oft üblich – alphabetisch geordnet, sondern nach praktischen Gesichtspunkten. Sie werden viele Beispiele finden, jedoch auch wertende, warnende, vor allem aber ermunternde Hinweise. Vorkenntnisse sind nicht nötig, ein paar Erfahrungen von CP/M oder im Umgang mit Standardsoftware würden das Verständnis der oft knappen Darstellung aber erleichtern.

Die ersten Schritte

Der erste Teil zeigt, wie das System gestartet wird und beschreibt das nicht ganz einfache DOS-Dateisystem. Die ersten Kommandos beziehen sich auf die Arbeit mit der Diskette. Schwerpunkt: Arbeit mit Unterverzeichnissen.

Historie

Als der größte Computerhersteller der Welt, die International Business Machines Corporation (IBM) 1980 nach einem Entwickler für das Betriebssystem (die Grundsoftware) ihres ersten 16-Bit-Personalcomputers suchte, kam dafür eigentlich nur Digital Research mit seinem Produkt CP/M-86, dem Nachfolger des erfolgreichsten kommerziellen Betriebssystems für 8-Bit-Technik, in Frage. Doch das Rennen machten Microsoft und sein MS-DOS (Microsoft Disc Operating System). Hierbei handelt es sich um ein von der Firma Seattle Computer Products unter dem Namen QDOS (Quick and Dirty Operating System) entwickeltes System mit Anteilen bei Microsoft (Arbeitsname DOS-86). Mit Markteinführung des IBM-PCs im August 1981 erwarb Microsoft alle Rechte, und gleichzeitig erschien auch PC-DOS, die identische IBM-Version von MS-DOS. Auch heute noch existieren beide Versionen nebeneinander; alle Nicht-IBM-PCs verwenden in der Regel MS-DOS.

Während seiner fast 10jährigen Geschichte unterlag MS-DOS einer ständigen Entwicklung, die besonders durch die Fortschritte bei der Hardware (z.B. Festplattenlaufwerke größerer Kapazität) hervorgerufen wurde. Leider scheint der ehemalige Name (Quick and Dirty = schnell und unsauber programmiert) dafür oftmals Programm gewesen zu sein.

Die erste DOS-Version (1.x) war CP/M noch sehr ähnlich. Mit der Version 2.x nahm es die Gestalt an, die uns auch heute noch vertraut ist. Dafür wurde eine Reihe erfolgreicher Prinzipien von Unix (einem in den Bell Telephone Laboratories, New Jersey, entwickelten portablen Betriebssystem mit modernen Konzepten) übernommen. Zu nennen wären das hierarchische Dateisystem, das in diesem Teil des Kurses noch zu besprechen sein wird), die Ein-/Ausgabe-Umleitung und die UNIX-ähnlichen Systemrufe, die heute fast ausschließlich verwendet werden. Viele der neueren Programme kommen auch noch mit dieser Version zurecht. Die Versionen 3.x brachten Funktionen für den Netzbetrieb (die Nutzung aus Turbo-Pascal heraus ist in MP 6/89 beschrieben), die Beachtung landesspezifischer Besonderheiten sowie die Unterstützung größerer Festplatten. Außerdem wurden einige vorhandene Systemkommandos erweitert.

Den gegenwärtigen Standard bilden die Versionen 3.2 und 3.3. Auf diese soll in diesem Beitrag auch Bezug genommen werden. Die inzwischen eingeführte Version 4.0 setzt sich nur zögerlich durch. Sie bringt uns neben der Nutzung großer Festplatten „in einem Stück" vor allem eine grafische Nutzeroberfläche. Statt der Eingabe von Kommandos werden Menüpunkte angeboten, die mit der Maus ausgewählt („angeklickt") werden.

Damit dürfte die Entwicklung von MS-DOS abgeschlossen sein. Grund dafür ist, daß aus Kompatibilitätsgründen – die alten CP/M-ähnlichen Funktionen werden immer noch unterstützt – die neuen Hardwaremöglichkeiten nicht genutzt werden können. Das bezieht sich besonders auf den Hauptspeicher oberhalb der 1-Mega-byte-Grenze und die neuen Möglichkeiten der Prozessoren (protected mode). Das Nachfolgesystem OS/2 beginnt sich langsam zu etablieren. Trotzdem wird DOS vorerst noch das am weitesten verbreitete Betriebssystem bleiben. Man spricht von 45 Millionen Installationen weltweit. Neben dem eigentlichen Betriebssystem hat sich noch eine Reihe von Erweiterungen entwikkelt, auf die in diesem Beitrag nicht eingegangen werden kann. Dazu gehören Nutzeroberflächen (wie MS-Windows und GEM), Netzschnittstellen und Möglichkeiten zur Nutzung eines größeren Speichers (wie Expanded Memory Specification). Im folgenden wird MS-DOS auch kurz als DOS bezeichnet.

Hardware

Herzstück des PCs ist ein Prozessor der Firma Intel. Beim IBM PC war es ein i8088, bei Kompatiblen meist ein i8086. Der einzige Unterschied besteht darin, daß der 8088 trotz interner 16-Bit-Architektur nach außen einen 8-Bit-Datenbus besitzt, so daß anfangs die alten Peripherieschaltkreise genutzt werden konnten. Sein Nachfolgemodell nannte IBM PC/XT (XT für eXtended Technology), dessen wichtigste Neuerung eine Festplatte war. Er – der Standard-PC unserer Tage – wurde inzwischen vom Modell PC/AT (AT für Advanced Technology) abgelöst. Er nutzt den Intel-Prozessor 80286 und im allgemeinen eine größere Festplatte. Damit sind vor allem Geschwindigkeitsvorteile verbunden. International vollzieht sich inzwischen auf breiter Front eine Ablösung durch den 80386, den ersten 32-Bit-Prozessor von Intel. Seine Stärken kann MS-DOS – wie schon erwähnt – nur unzureichend ausnutzen, deshalb bleibt er vor allem OS/2, UNIX und anderen leistungsfähigen Betriebssystemen vorbehalten.

Aber zurück zum Standard-PC. Vorwiegend ist er mit 640 KByte Hauptspeicher ausgerüstet. Das ist die obere Grenze für DOS. Viele Programme kommen auch mit wesentlich weniger Speicher aus, da die PCs der Vergangenheit zuerst mit 256, später mit 512 KByte ausgestattet waren. Doch der exzessive Umgang mit Speicherplatz hat inzwischen dazu geführt, daß für eine Reihe von Programmen die 640-KByte-Grenze schon Probleme mit sich bringt.

Start des Betriebssystems

Nach der Einführung wenden wir uns den ersten praktischen Schritten zu. Besitzt der PC eine Festplatte (und soviel sollten Sie über Ihren Computer schon wissen), so genügt zum Laden des Betriebssystems – der Fachmann sagt booten – das Einschalten. Wenn nicht, so wird der Anwender mit einer Ausschrift auf dem Bildschirm zum Einlegen der sogenannten Systemdiskette aufgefordert. Diese wird vom Hersteller mitgeliefert und ist als solche gekennzeichnet.

Laden Sie das System stets von der Diskette, so sollten Sie sich folgende Arbeitsschritte angewöhnen

- Einschalten
- Systemdiskette einlegen
- Reset-Taste betätigen.

Damit vermeiden Sie die eventuelle Zerstörung von Daten auf der Systemdiskette.

Anschließend fordert das Betriebssystem die Eingabe des aktuellen Tagesdatums und der Zeit durch folgende Ausschriften:

Beispiel:

```
Aktuelles Datum ist Die 11.07.1989
Neues Datum(mm.tt.jj):_
```

und

```
Uhrzeit: 12:06:45.18
Neue Uhrzeit:_
```

Vorausgesetzt, Sie arbeiten mit einer Betriebssystem-Version, die deutsche Ausschriften verwendet, ansonsten ergibt sich ein geringfügig anderes Bild. Das Datum ist wichtig für einige Systemfunktionen und sollte deshalb nicht durch Drücken der Enter-Taste übergangen werden. Hat Ihr Rechner keine batteriegepufferte Uhr, so wird er beim Einschalten den 01.01.1980, 00.00 Uhr – die Geburtsstunde des IBM PCs – anzeigen. Werden Sie nicht nach Datum und Zeit gefragt, so kann man dem abhelfen. Wie, das zeigt die nächste Folge.

Als nächstes erscheint die Copyrightausschrift des Betriebssystems und das Eingabebereitschaftszeichen (Laufwerksbuchstabe und Größerzeichen), Prompt genannt. Danach erscheint ein Unterstrich, der Cursor genannt wird. An dieser Position erscheint das nächste Zeichen, das über die Tastatur eingeben wird.

Beispiel:
Microsoft(R) MS-DOS(R) Version 3.30
(C)Copyright Microsoft Corp 1981–1987

A>_
Der Buchstabe symbolisiert das aktuelle Laufwerk, mit dem jetzt gearbeitet werden kann. Wurde das System von der Festplatte geladen, so erscheint der Buchstabe C (als logisch erstes Laufwerk der Festplatte). Nach dem Laden des Systems von der Diskette – wie in unserem Beispiel – wird das A angezeigt. In der Regel gilt folgende Zuordnung der Buchstaben zu den Laufwerken:

A erstes Diskettenlaufwerk
B zweites Diskettenlaufwerk
C Festplatte
D weiter logische Laufwerke

Es hängt von der vorhandenen Hardware ab (Größe der Festplatte, Anzahl der Diskettenlaufwerke), welche logischen Laufwerke Ihnen tatsächlich zur Verfügung stehen.
Einem physischen (wirklich vorhandenen) Laufwerk können mehrere logische (scheinbare) Laufwerke zugeordnet sein. So kann zum Beispiel das Laufwerk B: im DOS-Format und das gleiche Laufwerk als D: für ein CP/M-Format (800 KByte) genutzt werden (der Doppelpunkt hinter den Laufwerksbuchstaben kennzeichnet ihn als Gerät – im Gegensatz zu Dateien, die auch A oder B heißen könnten). Auch eine RAM-Disk (Simulation eines Laufwerkes im Hauptspeicher – oft als E: genutzt) ist möglich.
Das Umschalten auf ein anderes Laufwerk erfolgt mit dem Laufwerksbuchstaben und einem Doppelpunkt, sowie dem Betätigen der ENTER-Taste (im folgenden als <ENTER> dargestellt).
Zum Laufwerk C: geht es also mit der Eingabe von
c: <ENTER>
und zurück nach A: mit:
a: <ENTER>
Es ist gleichgültig, ob Sie große oder kleine Buchstaben verwenden. Das gilt für alle Kommandos im Betriebssystem. DOS wandelt alle Eingaben in große Buchstaben um. Auch die Dateinamen werden stets als Großbuchstaben angezeigt. Im folgenden werden hervorgehobene Kommandos mit Kleinbuchstaben fettgedruckt dargestellt.

Die Tastatur

Das Tastaturfeld ist nicht bei allen MS-DOS-fähigen Computern identisch, aber es gibt weitgehende Ähnlichkeit. Prinzipiell unterteilt sich das Tastenfeld in vier Bereiche: Buchstabentasten, Numerische Tastatur (separater Ziffernblock), Funktionstasten und Steuertasten.
Wichtige Funktionen haben die folgenden Tasten (das Einschließen in spitze Klammern kennzeichnet die jeweilige Taste):

<ENTER> Abschluß eines Kommandos
<SHIFT> Umschalten Groß-Kleinschreibung
<ESC> Löschen der Eingabezeile; in vielen Programmen Beenden des aktuellen Menüs
<ALT> Dient zur Eingabe erweiterter Codes.

Drücken Sie doch einmal die ALT-Taste und dann auf dem Ziffernblock rechts 42 – beim Loslassen der ALT-Taste erscheint ein *. Diese Möglichkeit wird noch einmal sehr wertvoll für Zeichen sein, die auf der Tastatur vermißt werden.

<CTRL> Ist gemeinsam mit anderen Tasten zu betätigen, z. B.
<CTRL> <H> ein Zeichen zurück, das gleiche wird auch mit dem Pfeil ← erreicht
<CTRL> <G> Löschen des aktuellen Zeichens (des Zeichens, auf dem der Kursor gerade steht)
<CTRL> <P> Zuschalten des Druckers, er druckt alles mit
<CTRL> <S> Anhalten des rollenden Bildschirms, dazu kommen wir noch
<PrtScr> druckt den aktuellen Bildschirm aus (Hardcopy)

Die Bedeutung der Funktionstasten (F1...F12) ist abhängig von Dienst- und Anwenderprogrammen. Es gibt leider keinen Standard für die Belegung der Funktionstasten durch Anwenderprogramme. Einige wenige Übereinstimmungen sind aber in den meisten Fällen vorhanden. Beispielsweise löst F1 bei den meisten Programmen eine Hilfsfunktion aus.
Im Betriebssystem werden nur zwei Tasten sinnvoll verwendet:
<F1> kopiert jeweils ein Zeichen des letzten Befehls in die Eingabezeile; so kann Zeichen für Zeichen der letzte Befehl hervorgeholt werden (z. B. bis zu dem Buchstaben, bei dem man sich verschrieben hat).
<F3> kopiert das letzte Kommando vollständig wieder in die Eingabezeile (z. B. wenn mehrmals der gleiche Befehl ausgeführt wird oder korrigiert werden soll).
Das gleichzeitige Betätigen der Tasten <CTRL> <ALT> <DEL> führt zum erneuten Laden des Betriebssystems (Warmstart). Es kann notwendig werden, wenn das Betriebssystem zum Beispiel durch unsaubere Programmabläufe seine Funktion nicht mehr erfüllt (Absturz).

Ein Blick auf die Systemdiskette

Auf jedem Datenträger ist ein Inhaltsverzeichnis (Directory) über alle vorhandenen Daten angelegt. Mit dem folgenden Befehl kann man sich dieses auf dem Bildschirm anzeigen lassen.
A>dir <ENTER>

Datentraeger in Laufwerk A ist Dos 3.30
Inhaltsverzeichnis von A:
COMMAND COM 2607 11.11.87 11.11
1 File(s) xxxxxx byte free

Bei der Eingabe von Befehlen können Groß- und Kleinbuchstaben gleichermaßen verwendet werden.
Der **DIR**-Befehl listet uns die Namen aller Dateien auf dem aktuellen Laufwerk. Ein Name besteht aus bis zu acht Zeichen. Diese dürfen beliebig sein, aber folgende Zeichen sind nicht erlaubt:
/ \ [ ] : ; , . = | " < > +
Warum der Doppelpunkt nicht erlaubt ist, wissen wir bereits. Auch die Zeichen ? und * haben eine andere Bedeutung, wie wir noch sehen werden. Ist der Dateiname kürzer als 8 Zeichen, so wird automatisch mit Leerzeichen aufgefüllt. Es folgen – durch einen Punkt getrennt – drei Zeichen, die den Typ der Datei kennzeichnen. Für die eigenen Dateien kann diese Erweiterung (Extention) frei (aber sinnfällig) gewählt werden, zum Beispiel

.TXT für allgemeine Textdateien
.DOC für Dokumentationen
.PAS für Pascal-Quelltexte
.ASM für Assembler-Quelltexte
usw.

Daneben gibt es Erweiterungen, die reserviert sind. Dazu gehören

.EXE (execute), ausführbare Programme im DOS-Hausformat
.COM (command), ausführbare Programme mit CP/M-Konventionen
.BAT (batch), Stapelprogramme, die einfache DOS-Befehle enthalten, wie sie auf Kommandoebene verwendet werden können
.$$$ temporäre Datei, wie sie z. B. Wordstar anlegt. Bei korrekter Arbeit dürfte sie nicht auf der Diskette vorzufinden sein, aber wenn während der Arbeit ein RESET erfolgt (Stromausfall oder RESET-Knopf) kann das schon mal vorkommen. Kann gelöscht werden.
.BAK Sicherungsdatei; die meisten Editoren bewahren die vorletzte Version unter diesem Namen zur Sicherheit auf.

Im allgemeinen ist an allen Stellen, wo ein Dateinamen gefordert wird, zusätzlich die Angabe eines Laufwerks zulässig.
Der DIR-Befehl präsentiert uns außer dem Dateiname noch die Größe der Datei, das Datum und die Uhrzeit der letzten Änderung. Beim Schreiben einer Datei (also auch beim Ändern) trägt DOS die Zeit automatisch ein.
Auf der Systemdiskette befinden sich neben den gezeigten noch zwei weitere Dateien, die DOS nicht anzeigt: IO.SYS (bzw. IBMBIO-.COM bei IBM) und MSDOS.SYS (bzw. IBM-DOS.COM). Sie besitzen die Attribute einer verborgenen Datei. Das hat nicht den Grund, diese Dateien vor irgend jemandem zu verstecken (schließlich haben wir unser Betriebssystem ja ehrlich gekauft), sondern soll den Bildschirm von ihnen frei und übersichtlich halten. Diese Dateien und die Datei COMMAND.COM gehören zum Betriebssystem. Sie müssen auf der Systemdiskette

```
Laden eines zweiten Kommandointerpreters

COMMAND [lw:] [pfad] [device] [/e:nnnn]
        [/p] [/d] [/f] [/c befehl]

lw:         Laufwerk
pfad        Zugriffspfad
device      legt das Standard-Ein-/Aus-
            gabegerät fest (Standard: CON)
/e:nnnn     Größe des Environments
            (Standard: 160 Byte
            Maximum: 32 KByte)
/p          installiert COMMAND resident,
            d.h. er kann nicht verlassen
            werden
/d          schaltet die Abfrage von Datum
            und Zeit ab
/f          unterdrückt der Rückfrage bei
            Diskettenfehlern
/c befehl   veranlaßt COMMAND zur Aus-
            führung eines Befehls und
            Rückkehr in das aufrufende
            Programm
```

(oder -platte) vorhanden sein. Während die ersten beiden Dateien (die den Betriebssystemkern enthalten) nur zum Systemstart benötigt werden, muß COMMAND.COM immer auf dem Startlaufwerk präsent sein. Diese Datei ist der Kommandointerpreter (oder auch Shell), der alle wichtigen Befehle der DOS-Kommandooberfläche – also zum Beispiel dir – enthält. Wird ein Programm geladen (beispielsweise WordStar oder dBase), so wird er nicht benötigt und deshalb überspeichert. Nach Verlassen des Programms lädt DOS ihn vom Startlaufwerk nach. Allein deshalb schon ist es ratsam, das System von der Festplatte zu laden. Für den Fall, daß nur ein Laufwerk zur Verfügung steht, erlaubt DOS einen Diskettenwechsel zum Nachladen des Kommandointerpreters.
COMMAND.COM ist ein Programm wie jedes andere. Wir können es mit

command <ENTER>

starten und würden uns jetzt im zweiten Kommandointerpreter befinden. Davon ist nichts zu merken (denn beide sind identisch). Verlassen wir ihn also mit

exit <ENTER>

wieder. Die Möglichkeit des Ladens eines zweiten Kommandointerpreters kann durchaus sinnvoll sein, etwa aus einem Programm heraus oder um ein komfortableres Modell zu nutzen.
Bleiben wir noch einen Moment beim DIR-Befehl. Soll nicht das aktuelle Laufwerk angezeigt werden, so kann ein Parameter angegeben werden, zum Beispiel

dir c: <ENTER>

für das Laufwerk C:. Leerzeichen sind stets dort nötig, wo sie in den Beispielen vorgegeben sind, in diesem Falle zwischen dir und c:.
Eine weitere Möglichkeit ist das Listen einer Gruppe von Files. Haben Sie zum Beispiel eine Textdatei geschrieben, deren Namen Sie vergessen haben (und warum sollte es Ihnen anders gehen als mir), und die Diskette ist voller Dateien: Hier wäre es möglich, sich alle Dateien vom Typ TXT auflisten zu lassen, und zwar mit

dir *.txt <ENTER>

Der Stern ist ein sogenannter Joker (oder Wildcard), ebenso wie das Fragezeichen. Ein Stern steht für einen beliebigen Namen oder eine beliebige Erweiterung (bzw. deren Rest), während das Fragezeichen für einen einzelnen beliebigen Buchstaben eingesetzt werden kann.

Einige Beispiele:

a*.*	alle Dateien, die mit a beginnen
pit.*	alle Dateien für Pit
M??er.txt	alle Textdateien für beliebig geschriebene Meiers
***.do?**	alle Dokumentationen, egal ob doc oder dok
.	alle Dateien

Die Joker haben nicht nur in Verbindung mit dem DIR-Befehl Bedeutung, sondern können überall verwendet werden, wo keine eindeutigen Dateibezeichnungen gefordert sind. Trotzdem darf man beim Löschen getrost etwas vorsichtig sein. Wer das Inhaltsverzeichnis seiner Diskette schwarz auf weiß mit nach Hause nehmen will, kann auch den Drucker zuschalten, der die Bildschirmanzeige mitdruckt.

Beispiel:

<CTRL> <P>
dir
<CTRL> <P>

Befinden sich sehr viele Dateien im Verzeichnis, so paßt die Anzeige nicht auf den Bildschirm. Dafür hält der DIR-Befehl zwei Schalter bereit:

dir/p Anzeige seitenweise (für page), nach 23 Zeilen wird angehalten und auf Tastendruck gewartet.
dir/w Ausgabe in fünf Spalten, dafür entfallen Dateigröße und Datum.

Das Dateiverzeichnis

Wer schon an 8-Bit-Technik gearbeitet hat und an dem heillosen Durcheinander auf den Disketten fast verzweifelt ist, der wird jetzt die Möglichkeit unter MS-DOS gerne nutzen, alle Dateien nach bestimmten Kriterien zu ordnen. Unter DOS gibt es nicht nur ein Verzeichnis, sondern ein Hauptverzeichnis (Rootdirectory) und beliebig viele Unterverzeichnisse (Subdirectories). In Bild 1 ist eine typische Verzeichnisstruktur dargestellt. Auf Disketten werden Sie nicht unbedingt Unterverzeichnisse finden, da der Platz begrenzt ist, aber auf der Festplatte sind sie ein Muß. Die Festplatte aus Bild 1 würde bei einem DIR folgendermaßen dargestellt werden:

```
Datentraeger in Laufwerk C ist ROGER
Inhaltsverzeichnis von C:\

TOOLS          <DIR>    15.01.89  08.15
TEXTE          <DIR>    18.06.89  10.00
DOS            <DIR>    19.10.89  14.00
COMMAND  COM   2607     28.02.89  17.30
         1 File(s) xxxxxx byte free
```

Angezeigt werden nur die Dateien im Hauptverzeichnis und dessen direkte Unterverzeichnisse. Diese werden durch <DIR> kenntlich gemacht. Um in das Verzeichnis TEXTE zu wechseln, wird der Befehl **CD** oder **CHDIR** (für Change Directory) verwendet:

cd texte <ENTER>
dir <ENTER>

```
Datentraeger in Laufwerk C ist ROGER
Inhaltsverzeichnis von C:\TEXTE
.          <DIR>   18.06.89  10.00
..         <DIR>   01.01.80  00.00
BRIEFE     <DIR>   15.01.89  08.15
BERICHTE   <DIR>   18.06.89  10.00
         0 File(s) xxxxxx byte free
```

Neben den beiden neuen Unterverzeichnissen (vergleiche Bild 1) werden noch zwei weitere Directories gefunden, die die Namen **.** und **..** haben. Dabei ist **.** das Verzeichnis, in dem wir uns gerade befinden (was an dieser Stelle ohne Belang ist) und **..** das übergeordnete Verzeichnis. So würde uns ein

cd ..

wieder ins Hauptverzeichnis zurückführen. Auf diese Weise kann man sich durch den gesamten Verzeichnisbaum bewegen. Um sich nicht von Ast zu Ast hangeln müssen, kann das Zielverzeichnis auch vollständig angegeben werden – unabhängig davon, wo man sich gerade befindet. Die Verzeichnisse werden durch ein \ (Backslash) getrennt; für das Wurzelverzeichnis steht nur ein \. Verwechseln Sie dieses Zeichen nicht mit dem Schrägstrich / (auch Slash genannt), wie er zum Beispiel für die Schalter des DIR-Befehls benutzt wird. Für das Abtrennen von Verzeichnissen kann dieser zwar in der Tat alternativ verwendet werden, aber zur Kennzeichnung des Hauptverzeichnisses ist nur der Backslash zulässig. Aufgrund der ungünstigen Dreifachbelegung der Tasten ist dieses Zeichen bei den meisten Tastaturen nur durch gleichzeitiges Betätigen der Tasten <CTRL> <ALT> und der entspechend gekennzeichneten Taste auf den Bildschirm zu bringen. Wie bereits beschrieben, können wir aber auch das Wissen ausnutzen, daß der Backslash als 92 (dezimal) kodiert wird. Durch Drücken der <ALT>-Taste und Eintippen von 92 auf dem rechten Ziffernblock. Beim Loslassen der <ALT>-Taste erscheint ein \.
Von jedem Punkt der Dateihierarchie kommt man mit

**cd **

ins Hauptverzeichnis und mit

cd \texte\briefe\liebe\susi

ins Verzeichnis Susi. Diese Angabe des Weges wird als *Pfad* bezeichnet. Grundsätzlich ist überall dort, wo ein Dateiname einzugeben ist, zusätzlich die Angabe eines Pfades möglich.

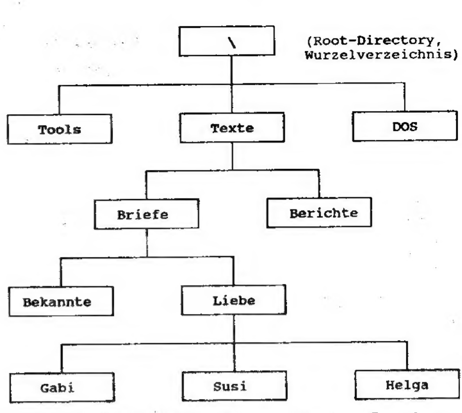
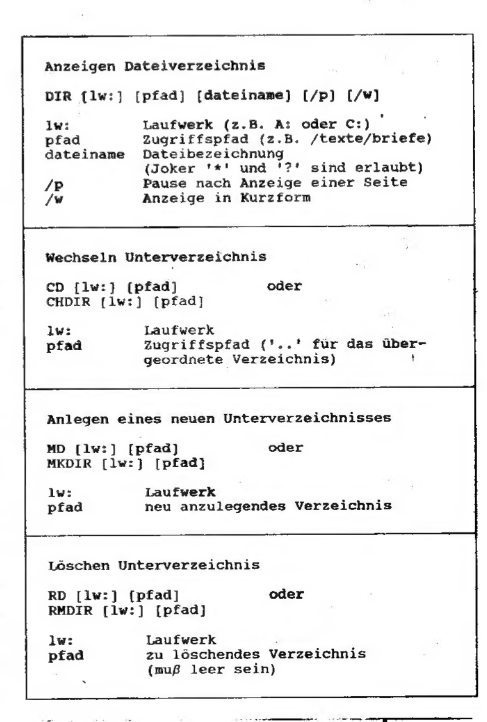


***Bild 1** Beispiel für ein hierarchisches Dateiverzeichnis*



```
Anzeigen Dateiverzeichnis

DIR [lw:] [pfad] [dateiname] [/p] [/w]

lw:         Laufwerk (z.B. A: oder C:)
pfad        Zugriffspfad (z.B. /texte/briefe)
dateiname   Dateibezeichnung
            (Joker '*' und '?' sind erlaubt)
/p          Pause nach Anzeige einer Seite
/w          Anzeige in Kurzform

Wechseln Unterverzeichnis

CD [lw:] [pfad]          oder
CHDIR [lw:] [pfad]

lw:         Laufwerk
pfad        Zugriffspfad ('..' für das über-
            geordnete Verzeichnis)

Anlegen eines neuen Unterverzeichnisses

MD [lw:] [pfad]          oder
MKDIR [lw:] [pfad]

lw:         Laufwerk
pfad        neu anzulegendes Verzeichnis

Löschen Unterverzeichnis

RD [lw:] [pfad]          oder
RMDIR [lw:] [pfad]

lw:         Laufwerk
pfad        zu löschendes Verzeichnis
            (muß leer sein)
```

Das Anlegen eines neuen Verzeichnisses ist mit dem Befehl **MD** oder **MKDIR** (für Make Directory) möglich. Um also die Briefe an unsere neue Errungenschaft richtig abzulegen, müssen wir uns mit
cd \texte\briefe\liebe
in das übergeordnete Verzeichnis bewegen und mit
md maria
ein neues Unterverzeichnis anlegen. Als Alternative steht die Angabe des gesamten Pfades:
md \texte\briefe\liebe\maria
Verzeichnisse können gelöscht werden, wenn sie keine Dateien und auch keine Unterverzeichnisse enthalten. Vom Löschen ebenfalls ausgenommen sind das Haupt- und das aktuelle Verzeichnis. Der Befehl dafür lautet **RD** oder **RMDIR** (für Remove Directory). Als Beispiel löschen wir das Verzeichnis, daß sich am ehesten entbehren läßt:
rd \texte\berichte

Weitere wichtige Befehle

Datei löschen
Mit dem **DEL**-Befehl können Dateien gelöscht werden. Als Parameter muß stets die Datei oder Dateigruppe angegeben werden, die gelöscht werden soll.
Beispiele:
del erfolg.txt
del \texte\briefe\liebe\susi\maerz.txt
del *.*
Die letzte Kommandozeile löscht alle Dateien des aktuellen Verzeichnisses. Bei der Verwendung der Joker beim Löschen ist Vorsicht geboten, oftmals kann man nicht übersehen, welche Dateien dazugehören. Die meisten Systeme akzeptieren statt **DEL** auch **ERASE**, was den CP/M-Nutzern entgegenkommt.

Datei umbenennen
Zum Umbenennen steht der Befehl **REN** oder **RENAME** zur Verfügung. Im Gegensatz zu CP/M erwartet DOS zuerst den alten und dann den neuen Namen.
Beispiel:
ren alt.txt neu.txt

Eingabe von Datum und Zeit
Zur Anzeige und zur Eingabe von Datum und Zeit dienen die Befehle DATE und TIME. Auf die Eingabe von **TIME** erscheint
Uhrzeit: 16:14:59.99
Neue Uhrzeit:
Die Anzeige kann mit <ENTER> quittiert werden, und wir wissen, daß in kürze Feierabend ist. Alternativ kann eine neue Zeit eingegeben werden, aber das sollten Sie ja schon am Anfang tun. Soll auf jeden Fall eine neue Zeit eingestellt werden, so kann das direkt geschehen:
time 16:30
Der **DATE**-Befehl funktioniert in gleicher Weise.

Kopieren

Einer der wichtigsten Befehle – wenn nicht gar der wichtigste überhaupt – ist der **COPY**-Befehl. Er wird in der Form
copy quelle ziel
verwendet und bietet eine Vielzahl von verschiedenen Anwendungsmöglichkeiten. Dabei stehen Quelle und Ziel für eine Datei oder eine Gruppe von Dateien (evtl. mit Laufwerk und Pfad) oder auch für ein Gerät. Als Ziel ist die Angabe eines Laufwerkes oder Verzeichnisses ausreichend.
Beispiel:
copy antrag.txt a: kopiert die Datei antrag.txt aus dem aktuellen Verzeichnis mit gleichem Namen nach a: Möglich ist auch das Kopieren innerhalb eines Laufwerkes in ein anderes Verzeichnis. Auf den Zielparameter kann ganz verzichtet werden, wenn es sich um das aktuelle Verzeichnis des aktuellen Laufwerkes handelt. Als Name der Zieldatei wird der Name der Quelldatei angenommen.
Beispiel:
copy a:\praemie.txt kopiert praemie.txt aus dem Wurzelverzeichnis des Laufwerkes a: ins aktuelle Verzeichnis des aktuellen Laufwerkes. Ist als Zielparameter auch ein Filename angegeben, so wird die Datei beim Kopieren umbenannt. Werden beim Zielnamen keine Joker verwendet, so muß auch der Quellname eindeutig sein.
Beispiel:
copy a:alt.txt b:neu.txt
Da es sich um einen Kopiervorgang handelt, bleibt die alte Datei selbstverständlich erhalten. Durch Umbenennen während des Kopiervorganges ist es möglich, eine Datei in dasselbe Verzeichnis zu kopieren. Auch die Joker können in vielfältiger Weise genutzt werden.
Beispiel:
copy *.txt a: kopiert alle Texdateien nach a:
Ein wichtiges Merkmal von DOS (auf das an dieser Stelle nicht näher eingegangen werden kann) ist, daß alle Geräte wie Textdateien behandelt werden, so daß es möglich ist, eine Datei auf den Bildschirm oder den Drucker zu kopieren. Zur besseren Unterscheidung werden die Geräte im folgenden groß geschrieben. Die wichtigsten DOS-Geräte sind:

CON	Konsole, beinhaltet Bildschirm und Tastatur
PRN	Drucker
COM1	erste serielle Schnittstelle

```
Datei(en) löschen

DEL [lw:] [pfad] [dateiname]          oder
ERASE [lw:] [pfad] [dateiname]

lw:         Laufwerk
pfad        Zugriffspfad
dateiname   Dateibezeichnung (Vorsicht bei
            Verwendung der Joker '*' und '?')
```

```
Datei(en) umbenennen

REN [lw:] [pfad]  datei_alt  datei_neu      oder
RENAME [lw:] [pfad]  datei_alt  datei_neu

lw:         Laufwerk
pfad        Zugriffspfad
            Laufwerk und Pfad werden nur
            für den alten Namen angegeben
datei_alt   alte Dateibezeichnung
datei_neu   neue Dateibezeichnung
```

```
Anzeige von Textdateien

TYPE [lw:] [pfad] dateiname

lw:         Laufwerk
pfad        Zugriffspfad
dateiname   Dateibezeichnung
```

```
Uhrzeit anzeigen/einstellen

TIME [hh[:mm[:ss[.xx]]]]

hh    Stunden     (0 .. 23)
mm    Minuten     (0 .. 59)
ss    Sekunden    (0 .. 59)
xx    Hundertstel (0 .. 99)
```

```
Datum anzeigen/einstellen

DATE [tt.mm.[jj]jj]

tt       Tag   (1 .. 31)
mm       Monat  (1 .. 12)
[jj]jj   Jahr (mit Jahrhundert
         oder ohne)
```

```
Anzeige MS-DOS Version

VER
```

```
Namen des Datenträgers anzeigen

VOL [lw:]

lw:    Laufwerk
```

```
Bildschirm löschen

CLS
```

```
Prüfen von geschriebenen Daten

VERIFY [on/off]

on     einschalten
off    ausschalten
```

```
Datei(en) kopieren

COPY [lw:] [pfad] dateiname [+[lw:]dateiname[+..]]
     [/a] [/b] [lw:] [pfad] [dateiname] [/a] [/b] [/v]

lw:         Laufwerk
pfad        Zugriffspfad
dateiname   Dateibezeichnung (vielfältige Verwendung
            der Joker möglich)
/a          Text-(ASCII-) Dateien
            (Ctrl-Z als Dateiendekennzeichen findet
            Beachtung)
/b          Programm-(Binär-) Dateien
            (Ctrl-Z wird nicht als Dateiendekennung
            betrachtet)
/v          (verify) Kontroll-Lesen nach dem Schreiben
```

Der Drucker kann nur als Ausgabegerät genutzt werden, die anderen Geräte zur Ein- und Ausgabe.
Einige Beispiele:
copy CON kleiner.txt Mini-Editor; alle Eingaben der Tastatur werden in die Datei geschrieben, Abschluß mit <CTRL> <Z> ENTER
copy CON PRN Schreibmaschine; die Ausgabe erfolgt auf den Drucker.
copy trans.txt COM1 kopiert den Text zur seriellen Schnittstelle, hoffentlich wird er dort auch aufgefangen.
Eine manchmal recht nützliche Möglichkeit des COPY-Befehls ist das Verketten von Dateien.
Beispiel:
copy eins.txt+zwei.txt
Soviel zu den vielfältigen Möglichkeiten des COPY-Befehls. Ein wenig Vorsicht ist geboten, da bereits existierende Zieldateien gleichen Namens ohne Warnung überschrieben werden. Ein Wunsch, den uns COPY nicht erfüllt, ist das Kopieren ganzer Verzeichnisse oder ganzer Disketten. Deshalb werden wir später Werkzeuge kennenlernen, die das unterstützen.

wird fortgesetzt

Einführung in MS-DOS

Teil 2

Wolfram Schulze, Uwe Schulze, Berlin

Nachdem wir in der ersten Folge (in MP 4/90) den Systemstart und die Arbeit im hierarchischen Dateisystem behandelt haben, steht diesmal das Anpassen (Konfigurieren) von DOS an die eigenen Anforderungen im Vordergrund. Außerdem befassen wir uns mit Aufbau, Formatierung und Prüfung von Disketten. Weiter im Programm: Die Veränderung der DOS-Bereitschaftsanzeige (Prompt).

Wir richten uns ein

Interne und externe Kommandos

Der erste Teil des Kurses schloß mit der Vorstellung einiger wichtiger Kommandos. DOS unterscheidet dabei zwischen internen und externen Kommandos. Als intern werden alle diejenigen Befehle bezeichnet, die vom Kommandointerpreter COMMAND.COM ausgeführt werden können. Es bedarf keiner zusätzlichen Programme auf der Diskette, denn sie befinden sich stets im Speicher. Alle bisher besprochenen Befehle gehörten dieser Gruppe an. Soll dagegen ein externes Kommando ausgeführt werden, so kann es schon einmal zu folgender Ausschrift kommen:

Falscher Befehl oder Dateiname

Nämlich dann, wenn die entsprechende Programmdatei nicht auf der Diskette (oder Festplatte) vorhanden ist, denn jedes externe Kommando steckt in einer eigenen Datei vom Typ COM (selten EXE).

Warum diese Einteilung in interne und externe Kommandos? Nun, die wichtigsten und am häufigsten verwendeten Befehle sollen stets zur Verfügung stehen, andererseits würden aber komplexe Kommandos zu viel Platz im Speicher einnehmen – zumal, wenn sie nur selten benötigt werden. Die Anzahl externer Kommandos hat sich von Version zu Version vergrößert, bzw. die alten wurden erweitert. Oftmals hilft also nur ein Blick ins Handbuch um festzustellen, ob ein bestimmter Befehl bei dieser Version bereits unterstützt wird. Auch ist Vorsicht beim Austausch der DOS-Systemkommandos geboten. Die Mehrzahl läuft nur unter der dazugehörigen Betriebssystemsversion. Das gilt auch für den Kommandointerpreter. Viele DOS-Kommandos haben durch das Auftauchen komfortabler Dienstprogramme ihre Bedeutung verloren, manche müssen sich der Konkurrenz modernerer Ausführungen erwehren (z. B. der Debugger DEBUG) und eines gehört mit Sicherheit ins Museum: der Kommandozeileneditor EDLIN.

Zwei wertvolle Kommandos wollen wir uns näher ansehen: **XCOPY** ist ein erweiterter Kopierbefehl, der ab DOS-Version 3.20 zur Verfügung steht. Die Ausgabe von Quell- und Zieldateien erfolgt wie bei COPY.

Beispiele:

xcopy *.txt a: /p Kopieren aller Textdateien aus dem aktuellen Verzeichnis nach A: mit Abfrage

xcopy *.* a: /s Kopieren aller Dateien und Unterverzeichnisse nach A:

Zu beachten ist, daß beim letzten Beispiel nicht notwendigerweise die gesamte Diskette kopiert wird. Befindet man sich schon in einem Unterverzeichnis, so werden Dateien und Verzeichnisse nur ab dieser Hierarchiestufe kopiert. Das Doppeln von Disketten kann mit

xcopy a: b: /s

erreicht werden. DOS stellt dafür aber eine bessere Möglichkeit zur Verfügung: Den Befehl **DISKCOPY**. Mit

diskcopy a: b:

können Sie ein Duplikat der in A: befindlichen Diskette in B: anlegen. Kopiert wird physisch (Spur für Spur), deshalb ist dieser Befehl schneller als XCOPY. Die Zieldiskette braucht nicht formatiert zu sein; sie erhält automatisch das Quellformat. Aber Achtung: Alle Daten der Zieldiskette werden irreparabel überschrieben. Hüten Sie sich davor, Quell- und Ziellaufwerk zu vertauschen!

Der Befehl gestattet es auch, auf Rechnern mit nur einem Laufwerk Kopien zu fertigen. Mit

diskcopy a:

dient A: als Quell- und Ziellaufwerk. In manchen Versionen muß es auch doppelt angegeben werden:

diskcopy a: a:

Nachdem DOS so viel wie möglich in den Speicher geladen hat, fordert es zum Diskettenwechsel auf, beschreibt die Zieldiskette usw. Ein weiterer Vorteil des DISKCOPY-Befehls besteht darin, daß der Diskettenname (Volumenname – Sie erinnern sich?) mitkopiert wird. Das kann bei Sicherheitskopien von gekaufter Software wichtig sein (und wird in den Handbüchern empfohlen), nämlich dann, wenn die Programme diesen Namen abfragen.

Es soll nicht verschwiegen werden, wann Sie DISKCOPY nicht verwenden können: Zwischen Disketten verschiedener Formate kann nicht kopiert werden. Und soll die starke Fragmentierung (Zersplitterung von Programmteilen) auf einer Diskette aufgehoben werden, so muß, wie oben gezeigt, auf XCOPY zurückgegriffen werden.

An dieser Stelle sei auch auf die Befehle **BACKUP** und **RESTORE** verwiesen, die für Sicherungskopien der Festplatte vorgesehen sind und mit denen es möglich ist, größere Dateien zu kopieren als der Zieldatenträger aufnehmen kann. Beachten Sie dabei bitte, daß beide Befehle unbedingt zur gleichen Version von DOS gehören sollten; ohne ein passendes RESTORE gibt es keine Möglichkeit, mit BACKUP gesicherte Daten je wieder zugänglich zu machen.

Anlegen neuer Disketten

Die gespeicherten Informationen sind auf den Disketten in Spuren abgelegt, die wiederum in Sektoren unterteilt sind. Die Sektoren sind bei MS-DOS einheitlich 512 Byte groß (während bei CP/M recht verschiedene Sektorgrößen Verwendung fanden: Zuerst 26 × 128 Byte, später 16 × 256 und 5 × 1024 Byte – ausführlich beschrieben in MP 2/1989). Auf Standarddisketten befinden sich gewöhnlich 9 Sektoren auf einer Spur, so daß das Format 9 × 512 Byte zustande kommt. Auch heute wird noch viel Software in diesem Format ausgeliefert (z. B. Büchern beiliegende Disketten). In der DDR findet überwiegend das zweite Format (720 KByte) Verwendung. Das *zweiseitige* Format bedeutet, daß das Laufwerk über zwei Schreib-/Leseköpfe verfügt und beide Seiten gleichermaßen genutzt werden. Noch mannigfaltiger ist das Angebot an Festplatten. Als Zylinder werden die jeweils übereinanderliegenden Spuren des Plattenstapels bezeichnet.

Vor der erstmaligen Benutzung eines Datenträgers muß dieser formatiert (oder auch initialisiert) werden. Das geschieht heute ausschließlich softwaremäßig (man spricht deshalb auch von Softsektorierung). Damit wird erst beim Formatiervorgang festgelegt, mit welchem Format ein Datenträger verwendet werden soll; durch erneutes Formatieren (allerdings bei Verlust aller Daten) kann das auch geändert werden. Der **FORMAT**-Befehl (ein externer Befehl) wird wie folgt verwendet:

format a:

zum Formatieren einer Diskette in Laufwerk A:. Soll eine Systemdiskette eingerichtet werden, so kann das mit

format a: /s

geschehen. Ohne weitere Schalter wird stets das Standardformat des jeweiligen Laufwerks angenommen. Ein beliebter Fehler bei der Verwendung dieses Befehls liegt in der falschen Laufwerksangabe. Also Vorsicht: Alle Daten werden zerstört. Am besten, sie entfernen die Disketten aus den anderen Laufwerken.

```
Diskette duplizieren

DISKCOPY lw1: lw2: [/1]

lw1: und lw2:   Laufwerksangaben
/1              nur einseitig
```

```
Erweiterter Kopierbefehl

XCOPY [lw:] [pfad] [datei] [lw:] [pfad] [datei]
      [/a] [/d:datum] [/e] [/m] [/p] [/s] [/v] [/w]

/a     nur Dateien, die seit dem letzten Backup
       geändert wurden
/e     auch leere Subdirectories werden kopiert
       (sofern /s gesetzt ist)
/m     wie /a, Archivbit wird zurückgesetzt
/s     Subdirectories werden mitkopiert
/v     Vergleichslesen wird eingeschaltet
/w     warten auf Tastendruck zum Disketten-
       wechsel
/d:datum  nur Dateien; die nach dem Datum ge-
          ändert wurden, werden kopiert
```

Beim Formatieren wird der Datenträger durch ein Kontrollesen geprüft, um so Materialfehler festzustellen.
Der Formatbefehl kann auch für Festplatten genutzt werden, allerdings sind vorher zwei weitere Schritte nötig: Die Vorformatierung mit einem speziellen Programm oder einem Sprung ins BIOS und die Aufteilung in Bereiche (Partitions) durch **FDISK**. Diese Arbeit sollte aber einem Experten überlassen werden. Auf jeden Fall kann der Inhalt einer Festplatte dadurch nicht versehentlich mit FORMAT zerstört werden.
Ist nicht mit dem Schalter /s formatiert worden, so können auch noch nachträglich mit
sys a:
die Systemdateien BIO.COM und DOS.COM übertragen werden. Der COPY-Befehl ist dafür nicht geeignet, da diese Dateien in einem festgelegten Bereich der Diskette stehen. Deshalb muß der Datenträger auch leer sein oder dieser Platz beim Formatieren mit dem /b-Schalter reserviert worden sein, da sonst wichtige Bereiche überschrieben werden. Anschließend ist noch COMMAND.COM mit einem beliebigen Kopierprogramm zu übertragen.
Jedem Datenträger kann ein Name zugeordnet werden. DOS stellt dafür den **LABEL**-Befehl zur Verfügung. Label bedeutet bekanntlich soviel wie Marke oder Etikett. Im EDV-Sprachgebrauch wird auch der Name von Datenträgern selbst als Label bezeichnet. Mit
label a: pit
erhält der Datenträger in Laufwerk A: den Namen PIT. Alle Kleinbuchstaben werden automatisch in Großbuchstaben umgewandelt. Der Name kann bis zu 11 Zeichen lang sein und – im Gegensatz zu Datei- und Verzeichnisnamen – Leerzeichen enthalten. Ausgeschlossen sind alle Zeichen, die auch für Dateinamen nicht verwendet werden dürfen.
Die Eingabe von
label
ohne einen Namen erzeugt folgende Ausschrift:

Diskette/Platte, Laufwerk A: hat keinen Namen
Name (max. 11 Zeichen) oder Eingabetaste
Besitzt der Datenträger bereits einen Namen, so wird dem Nutzer außerdem angeboten, diesen zu löschen. Der LABEL-Befehl kann jederzeit verwendet werden, also nicht nur solange der Datenträger leer ist. SYS und LABEL sind externe Befehle. Zur Erinnerung sei auch noch einmal auf den VOL-Befehl zum Anzeigen des so vergebenen Namens hingewiesen und, wie Ihnen sicher schon aufgefallen ist, wird er auch beim DIR-Befehl stets mit ausgegeben.

Überprüfen von Datenträgern

Ist ein Datenträger längere Zeit im Gebrauch, so können sich Inkonsistenzen (logische Widersprüche) im Dateisystem einstellen. Damit sind nicht mögliche Materialfehler gemeint – sie würden bereits beim Formatieren aufgespürt werden. Vielmehr handelt es sich um Fehler, die durch Stromausfall, Systemabsturz oder fehlerhafte Programme verursacht werden und die bewirken, daß Dateien nicht korrekt abgeschlossen sind. Die Folge ist, daß als belegt markierte Bereiche der Diskette (sie werden Cluster genannt) zu keiner Datei mehr gehören, also verloren sind, oder daß Bereiche mehreren Dateien zugeordnet sind, was den Inhalt verfälscht. DOS stellt das Programm **CHKDSK** zur Verfügung, das eine Prüfung von Datenträgern (Check Disk) vornimmt. Dazu werden die Verzeichniseinträge (Directories) herangezogen, sowie eine Tabelle, die auf jedem Datenträger vorhanden ist und die Belegung aller Cluster enthält. Diese heißt FAT (File Allocation Table) und ist im Augenblick für uns nicht weiter von Bedeutung.
chkdsk c:
überprüft den Datenträger des Laufwerks C: (i. a. eine Festplatte) und erzeugt folgendes Protokoll:

Diskette/Platte MO & PIT
erstellt 1 Jan 1980, 4.42
21309440 Byte Gesamtkapazität
81920 Byte in 6 geschuetzten Dateien
110592 Byte in 46 Verzeichnis(sen)
21053440 Byte in 1182 Benutzerdatei(en)
6144 Byte in fehlerhaften Sektoren
57344 Byte auf Diskette/Platte verfügbar

655360 Byte Gesamtspeicher
410336 Byte frei

Der obere Abschnitt bezieht sich auf die Festplatte (20 MByte) und der untere auf den Hauptspeicher (640 KByte). Zu den geschützten Dateien gehören die Systemdateien BIO.COM und DOS.COM. Wie bereits im Teil 1 beschrieben, besitzen diese die Attribute von geschützten und versteckten Dateien. Deshalb werden sie auch beim DIR nicht angezeigt. Zu den Dateiattributen mehr in einer späteren Folge.
Die Statistik zeigt uns ferner die Anzahl der insgesamt vorhandenen Unterverzeichnisse (46) und Dateien (1182) sowie den davon eingenommenen Platz. Auf der Festplatte sind nur noch 57344 Byte frei; wir werden uns also bald von einigen Daten trennen müssen.
Um eine Vorstellung von den Datenmengen zu vermitteln, werden als Faustregel für eine Seite Text 2 KByte veranschlagt. Mehr als 30 Seiten wären also auf der Festplatte nicht mehr unterzubringen.
Bei den hier angegebenen fehlerhaften Sektoren handelt es sich um Bereiche, die bereits beim Formatieren ausgemustert wurden. Die Platte wäre in diesem Falle also recht voll, aber in Ordnung. Kritisch sind Fehler, wenn CHKDSK feststellt, daß Cluster zu keiner Datei gehören oder doppelt zugeordnet sind. In diesem Falle sollte der /f-Schalter angegeben werden; nach Rückfrage versucht CHKDSK diesen Fehler zu beheben. In den meisten Fällen ist eine Reparatur möglich, wenn nicht, muß darüber nachgedacht werden, alle Dateien zu retten und den Datenträger neu zu formatieren.
CHKDSK kann auch dazu benutzt werden festzustellen, wie stark einzelne Dateien fragmentiert sind. Die Verwaltung der Dateien führt im Laufe der Zeit dazu, daß sie nicht mehr zusammenhängend abgespeichert sind. Sie werden „zerstückelt" oder – fragmentiert. Die Folge sind längere Ladezeiten, verursacht durch das Zusammensuchen der einzelnen Teile auf dem Datenträger. Auskunft darüber gibt CHKDSK mit Angabe eines Filenamens, also
chkdsk turbo.exe
zum Feststellen der Fragmentierung des Turbo-Pascal-Compilers.
Selbstverständlich können auch die Joker für mehrere Dateien Verwendung finden. Ein Aufheben der Fragmentierung kann nur durch Umkopieren erreicht werden (siehe Diskussion zu XCOPY) oder durch spezielle Verdichtungsprogramme (z. B. COMPRESS).

AUTOEXEC.BAT

Stets wiederkehrende Befehlsfolgen können im DOS in speziellen Kommandodateien zusammengefaßt werden. Sie tragen den Typ BAT (von Batch) und werden deshalb auch Batch-Dateien oder Stapeldateien (aus dem Großrechner-Vokabular) genannt. An einem

```
Formatieren von Datenträgern

FORMAT lw: [/1] [/4] [/8] [/s] [/v] [/b]

lw:     Laufwerk
/1      formatiert nur einseitig
/4      formatiert 360 KByte-Disketten auf
        HD-Laufwerk
/8      formatiert 8 Sektoren/Spur (Standard: 9)
/s      die drei Systemdateien werden automatisch
        mit übertragen
/v      nach dem Formatieren wird nach einem
        Diskettennamen (Volumelabel) gefragt
/b      der Platz für das Betriebssystem wird
        reserviert, auch wenn inzwischen Dateien
        auf die Diskette kopiert werden
```

```
Übertragen des Betriebssystems

SYS lw:

Es werden nur die beiden Dateien des BIOS und BDOS
übertragen; für COMMAND.COM muß ein normaler Kopier-
befehl verwendet werden. Voraussetzung ist, daß der
Datenträger leer ist oder mit dem /b - Schalter for-
matiert wurde.
```

```
Namen für einen Datenträger vergeben

LABEL [lw:] [name]

lw:    Laufwerk
name   Zeichenkette (bis zu 11 Zeichen)
```

```
Überprüfen eines Datenträgers

CHKDSK [lw:] [pfad] [datei] [/f] [/v]

lw:     Laufwerk
pfad    Zugriffspfad
datei   Dateiname
/f      nach Rückfrage ist eine Fehlerkorrektur
        möglich
/v      protokolliert, an welcher Datei jeweils
        gerade gearbeitet wird
```

kleinen Beispiel sei das einmal demonstriert. Wir benutzen als Editor den COPY-Befehl, wie in Teil 1 erläutert. Sie können aber auch Ihren Hauseditor benutzen. Folgende Kommandos dienen zum Festlegen eines kleinen Testfiles:

copy con test.bat
cls
ver
<CTRL> <Z>

Abgeschlossen wird mit der Tastenkombination <CTRL> <Z> (<Strg> <Z>). Die damit erzeugte Datei kann man sich mit

type test.bat

ansehen, und beim Aufruf mit dem Dateinamen

test

werden die enthaltenen Befehle automatisch ausgeführt. In diesem Falle wird der Bildschirm gelöscht und die DOS-Version angezeigt.

Auf diese Weise können beliebige Kommandos automatisch und wiederholt abgearbeitet werden. In der nächsten Folge des Kurses geben wir eine ausführliche Darstellung der Möglichkeiten der Batch-Programmierung sowie nützliche Tips und Tricks. An dieser Stelle soll nur eine ganz besondere Kommandodatei besprochen werden. Beim Systemstart sucht der Kommandointerpreter nach einer Datei mit dem Namen AUTOEXEC.BAT. Findet er sie nicht, so werden Datum und Zeit abgefragt, ist sie vorhanden, werden die enthaltenen Befehle abgearbeitet. Damit hat der Nutzer ein Mittel in der Hand, beim Start stets wiederkehrende Befehlsfolgen ausführen zu lassen. Eine typische AUTOEXEC.BAT (auch Start Up File genannt) könnte folgendes Aussehen haben:

echo off
break on
verify on
ver
prompt pg
path c:\dos; c:\tools
append c:\dos; c:\tools
keyb gr
date
time

Mit **echo off** wird die Protokollierfunktion (auch Echofunktion genannt) ausgeschaltet. Die folgenden Befehlszeilen erscheinen somit nicht mehr auf dem Bildschirm – wohl aber die durch sie erzeugten Ausgaben. Diese Anweisung dient nur der Erhöhung der Übersichtlichkeit. Am Ende der Batch-Datei schaltet sich das Echo automatisch wieder ein; wird der Befehl direkt von DOS aus gegeben, so muß es explizit mit **echo on** wieder eingeschaltet werden. Wird echo ohne Parameter eingegeben, zeigt der Bildschirm die gegenwärtige Schalterstellung an. Außerdem kann der Befehl echo dazu benutzt werden, um Nachrichten auf den Bildschirm auszugeben; eine Option, die nur in Stapeldateien einen Sinn gibt. Versuchen Sie es einmal mit

echo alles roger

Mit dem **BREAK**-Befehl (interner Befehl) wird festgelegt, wann die Tastenkombination <CTRL> <BREAK> (bei den meisten Rechnern auch <CTRL> <C>) Zum Programmabbruch führen soll. Mit break off kann die Überprüfung abgeschaltet werden (was auch Standard ist). Genaugenommen ist die Auswertung der Unterbrechungstasten nicht völlig unterbunden; bei einigen Ein- und Ausgabeoperationen sind sie trotzdem noch wirksam. Das Einschalten wird mit break on erreicht. Diese Option ist besonders für Programmierer geeignet, die eventuell vorkommende Endlosschleifen abbrechen müssen. Voraussetzung ist aber, daß DOS-Systemrufe enthalten sind, denn nur dort findet eine Auswertung der Tastenkombination <CTRL> <BREAK> (<Strg> <Untbr>) statt. Beispielsweise läßt sich so auch während der Diskettenarbeit ein Abbruch erzielen (z. B. während FORMAT).

Dem PROMPT-Befehl ist am Ende dieses Kurses ein ausführlicher Abschnitt gewidmet. Kommen wir deshalb gleich zum **PATH**-Befehl. Dieser wird in kaum einer AUTOEXEC.BAT fehlen. Er legt fest, wo DOS nach Programmen suchen soll, die sich nicht im aktuellen Verzeichnis befinden. Beispielsweise möchte man Wordstar oder ein anderes Textverarbeitungsprogramm seines Vertrauens von jedem Textverzeichnis aus aufrufen können, aber (aus Platzgründen) nicht in jedes Verzeichnis kopieren. Gleiches gilt für die externen DOS-Befehle. Wie Sie sich erinnern werden, handelt es sich dabei um eigene Programme. Wollen Sie also den XCOPY- oder den FORMAT-Befehl aufrufen, so muß DOS wissen, wo er zu finden ist – denn im aktuellen Verzeichnis ist er mit großer Wahrscheinlichkeit gerade nicht.

Zuerst wird stets im aktuellen Verzeichnis gesucht, dann der Reihe nach in den Verzeichnissen, die der PATH-Befehl angibt. Im Falle der hier als Beispiel gezeigten AUTOEXEC.BAT würden die Verzeichnisse DOS und TOOLS durchmustert werden. Erst wenn das Programm in allen diesen Pfaden nicht gefunden wird, erscheint

Falscher Befehl oder Dateiname

Es kann auch auf verschiedenen Laufwerken gesucht werden, beispielsweise in den Wurzelverzeichnissen von A: und C: mit

path c:\; a

Üblicherweise wird man aber nur auf der Festplatte oder der Systemdiskette suchen lassen, um nicht auf leere Laufwerke oder falsch eingelegte Disketten zuzugreifen. Mit

path

ohne Parameter wird der aktuelle Pfad angezeigt, und mit

path;

wird der Pfad gelöscht.

Der PATH-Befehl ist ein sehr wertvoller Befehl, es sei aber auch nicht verschwiegen, daß sich bei einem langen PATH die Ladezeiten verlängern. Nicht zu vergessen ist, daß mit PATH nur ein Suchpfad für ausführbare Programme (der Typen COM, EXE und BAT) festgelegt werden kann. Insbesondere mit nachladbaren Programmteilen (sogenannte Overlays), die nicht gefunden wurden, gibt es Probleme. Deshalb steht ab Version 3.20 der zusätzliche Befehl **APPEND** zur Verfügung. Im oben gezeigten Beispiel der AUTOEXEC.BAT sind beide Pfade identisch, es ist aber auch möglich, zwei verschiedene Suchpfade zu definieren, wenn Programmdateien und Nicht-Programmdateien in gesonderten Verzeichnissen stehen. Die Verwendung des APPEND-Befehls erfolgt analog dem PATH-Befehl. Ohne Parameter aufgerufen, zeigt er den aktuellen Suchpfad an, und mit

append;

wird der Pfad gelöscht. Es sei noch darauf verwiesen, daß PATH ein interner Befehl ist, während es sich bei APPEND um einen externen handelt.

Mit **KEYB GR** wird der deutsche Tastaturtreiber installiert; Standard ist die amerikanische Belegung. Sie werden das zuerst an der Lage des Doppelpunktes bemerken, der zum Umstellen der Laufwerke benötigt wird. Die amerikanische Tastatur kann mit der Tastenkombination <CTRL> <ALT> <F1> jederzeit wieder aktiviert werden, mit <CTRL> <ALT> <F2> gelangt man zurück zur deutschen Belegung. Mit einer anderen Landeskennung können auch weitere Tastaturtreiber anderer Länder geladen werden; nur die jeweils letzten beiden sind aber aktivierbar. Vor der DOS-Version 3.30 war die Handhabung für landessprachliche Tastaturtreiber etwas anders. Hier existierte für jedes Land ein eigener Treiber. Der entsprechende Befehl heißt

keybgr

Äußerlich besteht der Unterschied nur im fehlenden Leerzeichen, in Wirklichkeit existiert hier für jedes Land ein eigenes Kommando, während ab DOS 3.30 die Landeskennung nur ein Parameter für KEYB ist, der die jeweiligen Zeichensätze aus einer Datei holt.

Da die Datei AUTOEXEC.BAT eine ganz normale Batch-Datei ist, die sich nur durch ihren reservierten Namen auszeichnet, kann sie auch jederzeit durch Aufruf mit dem Namen

autoexec

gestartet werden. Umgekehrt können alle Befehle, die in ihr enthalten sind, auch einzeln auf Kommandoebene angewendet werden bzw. können beliebige andere DOS-Kommandos oder Programme in die AUTOEXEC.BAT geschrieben werden. Sollen beim Systemstart alle .BAK-Dateien gelöscht werden, so könnte das mit

del *.bak

in der AUTOEXEC.BAT geschehen. Beliebt ist die Möglichkeit, beim Systemstart gleich ein Programm aufzurufen. Wenn sie stets mit dBase arbeiten, so können sie als letzten Befehl

dbase

einfügen und gelangen beim Einschalten automatisch in das Anwenderprogramm. Zur Unterstützung unbedarfter Nutzer ist dies sehr hilfreich, da keine DOS-Kenntnisse nötig sind. Nach dem Studium unseres Kurses wird das für Sie aber überflüssig sein.

Die Konfigurationsdatei CONFIG.SYS

Beim Systemstart sucht DOS nach einer Datei, die Angaben zur Konfiguration enthält. Ihr Name ist CONFIG.SYS, und sie legt vor allem fest, welche Treiber (Gerätesteuerprogramme) geladen und welche Pufferbereiche reserviert werden sollen. Die Angaben in dieser Datei sind – im Gegensatz zu allem, was wir bisher besprochen haben – keine Befehle, die ausgeführt werden können. Deshalb werden sie im folgenden als Parameter bezeichnet. Eine typische CONFIG.SYS könnte folgendes Aussehen haben:

files = 20
buffers = 20

```
Protokollierfunktion
ECHO [on/off/text]

on      einschalten
off     abschalten
text    Ausgabe des Textes auf
        den Bildschirm
```

```
Prüfung auf Abbruch mit <CTRL> <BREAK>
BREAK [on/off]

on       Überprüfung bei jedem DOS-Ruf
off      Überprüfung nur bei Ein-/Ausgabe-
         Funktionen
```

```
Suchpfad für Programm-Dateien festlegen

PATH [=] [lw:] [pfad] [;[lw:][pfad]...]
bzw.
PATH;           zum Löschen des Pfades

=        keine Bedeutung, Benutzung wahlweise
lw:      Laufwerk
pfad     Suchpfad
```

```
Suchpfad für Nicht-Programm-Dateien

APPEND [lw:] [pfad] [;[lw:][pfad]...] [/x] [/e]
bzw.
APPEND;         zum Löschen des Pfades

lw:      Laufwerk
pfad     Suchpfad
/x       erweitert die Suche auch auf Programm-
         dateien
/e       speichert den Suchweg im Environment
         (wie bei Path)
```

```
Anpassung nationaler Tastaturen

KEYB [xx] [,nnn] [datei]

xx       Landeskennung: GR für Germany
nnn      Nummer des Zeichensatzes
datei    Name der Datei, die Tastatur-
         Spezifikation enthält
```

```
K o n f i g u r a t i o n s b e f e h l e
              in CONFIG.SYS

BREAK = ON          Prüfung auf Abbruch
        OFF         mit <CTRL> <BREAK>
                    ein-/ausschalten
                    (Standard: OFF)

BUFFERS = n         Anzahl der Puffer für Fest-
                    plattenzugriffe (1..99)

COUNTRY = n         Einstellung landesspezifischer
                    Schreibweisen (Datum, Zeit)
                    n  =  001    USA  (Standard)
                    n  =  049    Bundesrepublik
                                 Deutschland
                    (entspricht der Vorwahl im
                     internationalen Telephon-
                     verkehr)

DEVICE = datei      Installation von Geräte-
                    treibern

FILES = n           maximale Anzahl gleichzeitig
                    offener Dateien   (8..255)

FCBS = m,n          maximale Anzahl gleichzeitig
                    offener File Control Blocks
                    (1..255) (CP/M-kompatibel)
                    m  Anzahl
                    n  Anzahl von FCBs, die nicht
                       geschlossen werden sollen,
                       wenn m überschritten ist

SHELL = [lw:] [pfad] [datei]          oder
SHELL = [lw:] [pfad] command.com [parameter]

                    Festlegung des Kommandointer-
                    preters
                    Entweder wird ein eigener
                    Kommandointerpreter installiert
                    oder COMMAND.COM mit Parametern
                   (siehe Beschreibung COMMAND.COM)
```

country = 049
device = \dos\ansi.sys
lastdrive = f
Mit **FILES** kann die Anzahl der gleichzeitig geöffneten Dateien festgelegt werden. Standard sind 8, höchstens können 255 angegeben werden. Zu beachten ist, daß in dieser Zahl auch die 5 Standard-Eingabe- und -Ausgabekanäle (Konsole, Drucker, Fehlerausgabe) enthalten sind, so daß bei einer Zahl von 8 nur noch drei nutzereigene Dateien verbleiben. Die meisten großen Programmpakete (z.B. dBase) geben in der Installationsanleitung an, auf welche Zahl FILES gesetzt werden soll. Es sei noch kurz erwähnt, daß Files nur für die neueren – heute fast ausschließlich genutzten – Unix-ähnlichen DOS-Funktionen zuständig ist. Für ältere Programme muß die Einstellung mit FCBS erfolgen. Weitere Erklärungen würden an dieser Stelle zu weit führen.
BUFFERS legt die Anzahl interner Puffer (a 512 Byte) fest, die bewirken, daß weniger Disketten- (bzw. Platten-)Zugriffe nötig sind, weil jeweils eine größere Datenmenge als ein Sektor in den Speicher gelesen wird. Der Sektor ist die kleinste Einheit, die mit einem Mal gelesen oder geschrieben werden kann. Möglich sind Werte zwischen 1 und 89, Standard ist 2, und empfohlen werden Werte zwischen 10 und 20, da bei größeren Zahlen nicht nur zu viel Speicherplatz verloren geht, sondern die Zugriffe auch wieder langsamer werden (aufgrund der langen Suche in den Puffern).
Zum Einstellen landesüblicher Schreibweisen dient der **COUNTRY**-Parameter. So schreiben die Amerikaner das Datum mit Schrägstrichen (06/18/90), die Europäer mit Punkt (18.06.90). Unterschiedlich sind auch die Zeitangaben (2:00 pm bzw. 14:00) und die Abtrennung der Tausender bei Zahlenkolonnen (mit Punkt bzw. Komma). Standard ist die amerikanische Einstellung (Kennung 001). Mit
CONTRY = 049
wird die deutsche Schreibweise eingestellt. Die Länderkennungen entsprechen der internationalen Telefonvorwahl (049 für Bundesrepublik Deutschland). Manchmal wird auch die 037 (für DDR) angeboten. Beachten Sie bitte, daß die Anwendung von COUNTRY keinen Einfluß auf die Tastaturbelegung und die Benutzung der Umlaute hat.
Für alle Geräte (Konsole, Drucker, Laufwerke, serielle Schnittstelle) lädt DOS beim Start eigene Steuerprogramme (Treiber, englisch: device driver). Der Nutzer hat die Möglichkeit, die Installation zusätzlicher Treiber zu veranlassen. Dafür wird der **DEVICE**-Parameter verwendet. Ziel ist entweder der Anschluß zusätzlicher Geräte oder das Ersetzen vorhandener Treiber durch einen eigenen (besseren). Über diese Problematik informieren auch MP 11/89, Seite 328, und 12/89, Seite 359. Wichtige zusätzliche Treiber sind ANSI.SYS (im nächsten Teil beschrieben), MOUSE.SYS zur Bedienung der Maus und VDISK.SYS zum Anlegen einer RAM-Disk. Der DEVICE-Parameter kann mehrmals in der CONFIG.SYS erscheinen, je nachdem, wie viele zusätzliche Treiber eingebunden werden sollen. Erfordern die Treiber zusätzliche Parameter, so können sie in der Zeile mit angegeben werden, beispielsweise
device = vdisk.sys 128 512 24
für eine RAM-Disk mit 128 KByte Kapazität, 512 Byte Sektorlänge und 24 Hauptverzeichniseinträgen. Welche Parameter für jeden einzelnen Treiber möglich sind, muß den jeweiligen Handbüchern entnommen werden. Der Eintrag **LASTDRIVE =F** legt fest, daß F: das letzte mögliche logische Laufwerk sein soll. Standardeinstellung ist E:. Dieser Parameter steht im Zusammenhang mit der Möglichkeit, ein physisches Laufwerk mit verschiedenen Formaten als logische Laufwerke zu nutzen sowie mit dem SUBST-Befehl (wird in einem späteren Teil besprochen) und Laufwerksumweisungen im Netzwerk. Für jedes Laufwerk wird Speicherplatz für die Verwaltung durch DOS reserviert, vergeben Sie also nicht mehr Platz als nötig. Werden weniger Laufwerke eingetragen, als tatsächlich vorhanden, so wird die Angabe ignoriert.
Zwei Parameter sind nicht verwendet worden: Mit **BREAK** kann die Prüfung auf <CTRL> <BREAK> zum Programmabbruch ein- und ausgeschaltet werden (exakt die gleiche Wirkung, wie das gleichnamige DOS-Kommando), und mit **SHELL** wird ein eigener Kommandointerpreter festgelegt, sofern nicht COMMAND.COM verwendet werden soll. Alternativ lassen sich damit auch Änderungen in der Installation von COMMAND.COM bewirken, etwa ein Verzeichnis, aus dem er stets geladen werden soll, oder die Größe der Programmumgebung.
Nicht näher beschrieben werden in diesem Abschnitt die Konfigurationsparameter STACKS (zur Festlegung der Größe der Interruptstacks) und DRIVEPARM (zur Änderung der Laufwerksparameter). Gegebenenfalls müssen Sie im Handbuch nachschlagen. Ab Version 4.00 kommen INSTALL (zum Installieren residenter Programme), REM (für Kommentare innerhalb von CONFIG.SYS) und SWITCHES (zur Unterdrückung erweiterter Tastenfunktionen) hinzu. Es ist auch jetzt schon möglich, mit REM Kommentare einzufügen, damit wird aber eine Fehlermeldung verursacht. REM (von Remark) erinnert den Nutzer an seine Absicht, einen Kommentar eingefügt zu haben.
Werden Eintragungen in der CONFIG.SYS geändert, so muß das System auf jeden Fall neu gestartet werden (<CTRL> <ALT> <DEL> oder Reset-Taste), da die Konfigurationsdatei nur zu diesem Zeitpunkt beachtet wird. Auch ein Entfernen der so installierten Treiber ist grundsätzlich nur durch den Neustart möglich. Da dies oft falsch verstanden wird, sei noch einmal darauf hingewiesen, daß die Angaben in der CONFIG.SYS keine ausführbaren Befehle sind und deshalb nicht auf Kommandoebene gestartet werden können.

wird fortgesetzt

Einführung in MS-DOS

Teil 3

Wolfram Schulze, Uwe Schulze, Berlin

Der dritte Teil des Kurses widmet sich dem Prompt-Befehl, zeigt, was es mit dem ANSI-Treiber auf sich hat und stellt die kleine, aber leistungsfähige Kommandosprache von DOS vor.

Nutzung des Prompt-Befehls

Als Prompt wird das Bereitschaftszeichen von DOS bezeichnet. Es zeigt das aktuelle Laufwerk, meistens den aktuellen Pfad und ein Größerzeichen. Dem Nutzer ist es überlassen, den Prompt nach eigenem Belieben zu verändern. Versuchen Sie es doch einmal mit

prompt alles roger

und drücken Sie ein paarmal <ENTER>, um zu sehen, was passiert. Ganz lustig ist das ja, aber nicht allzu hilfreich, da wir nicht mehr wissen, auf welchem Laufwerk wir uns befinden. Mit

prompt

ohne Parameter läßt sich wieder die Standardanzeige einstellen. Außer beliebigem Text kann der Prompt aber noch eine Reihe von Systeminformationen anzeigen. Dafür sind einige Sonderzeichen reserviert. Sie bestehen jeweils aus einem Dollarzeichen ($) und einem Buchstaben. Im folgenden sind die möglichen Kombinationen aufgelistet:

$b das Zeichen |
$d das Datum
$e das Zeichen Escape (1BH)
$h letztes Zeichen löschen
$g das Zeichen >
$l das Zeichen <
$n das aktuelle Laufwerk
$p das aktuelle Verzeichnis
$q das Zeichen =
$s das Leerzeichen
$t die Uhrzeit
$v die DOS-Version
$_ neue Zeile (CR/LF)
$$ das Zeichen $

Der Standardprompt, den man erhält, wenn man PROMPT ohne Parameter aufruft, zeigt das Laufwerk und das Größerzeichen. Er entspricht damit der Einstellung prompt ng. Allgemein üblich ist es aber, den aktuellen Pfad anzuzeigen. Dazu dient die Anweisung

prompt pg

wie sie auch in unserer Datei AUTOEXEC-.BAT eingetragen ist. Auf diese Art und Weise lassen sich beliebig komfortable Prompts einrichten, etwa

prompt Zeit:t_Datum$d$_pg

Vergessen Sie über das Experimentieren aber nicht, daß der Prompt nicht nur nett aussehen, sondern auch zweckmäßig sein soll.

Der Prompt-Befehl und ANSI.SYS

Die Bildschirmsteuerung unter MS-DOS ist nicht besonders komfortabel. Außer dem Piepton (Bell), dem Wagenrücklauf (Carriage Return/Linefeed) und dem Löschen des letzten Zeichens (Backspace) werden keine Steuerzeichen interpretiert. Deshalb kann ein zusätzliches Gerätesteuerprogramm (Treiber, englisch Device Driver) verwendet werden, das weitere Steuerbefehle verarbeitet. Man hält sich damit an einen Normvorschlag für Terminal-Steuerbefehle (Normvorschrift X 3.64: Additional Controls for Use with ASCII) der amerikanischen Standardisierungsbehörde ANSI (American National Standards Institute). Deshalb wird der Treiber auch kurz ANSI-Treiber genannt und wir finden ihn auf der Systemdiskette unter dem Namen ANSI.SYS. Um seine Dienste in Anspruch zu nehmen, muß er mit

device=ansi.sys

in die Konfigurationsdatei CONFIG.SYS (siehe Teil 2) eingetragen werden. Steht der ANSI-Treiber nicht im Hauptverzeichnis, so kann auch sein Pfad angegeben werden, beispielsweise

device=\dos\ansi.sys

Anschließend muß das System neu gestartet werden (<CTRL> <ALT> <DEL>, <Strg> <ALT> <Entf> oder Reset-Taste); nur so sind neue Treiber zu installieren. Ab sofort untersucht DOS alle zum Bildschirm ausgegebenen Zeichen auf Steuerfolgen. Da fast alle Tasten in ihrer eigentlichen Bedeutung (Buchstaben, Zahlen, Sonderzeichen) gebraucht werden, stehen nicht genug Steuerzeichen zur Verfügung. Deshalb werden Folgen von Zeichen verwendet, die stets durch das Zeichen Escape (ASCII-Code 1B hex. = 27 dezimal) und die öffnende eckige Klammer [(ASCII-Code 5B hex. = 91 dezimal) eingeleitet werden. Daran erkennt der ANSI-Treiber, daß die folgenden Zeichen nicht als Zahlen, Buchstaben etc., sondern als Steuerfolge zu interpretieren sind. Das einleitende Escape steht auch für die Bezeichnung Escape-Sequenzen (Sequenz (engl.) = Folge, Ordnung). Tafel 1 gibt einen Überblick über die möglichen Befehlsfolgen. Sie müssen nur noch auf den Bildschirm ausgegeben werden und bewirken das entsprechende Kommando. In einer Programmiersprache können die normalen Ausgabeanweisungen verwendet werden, also *write* in Pascal oder *printf* in C. (später ein Beispiel dazu).

Aber auch im Betriebssystem gibt es Kommandos, die Ausgaben auf den Bildschirm bewirken. Erinnern Sie sich bitte an die letzte Kurs-Folge. Mit

echo Dies ist eine wunderschöne Ausgabe

erhalten Sie eben diese Ausgabe auf den Bildschirm. Leider läßt sich das Escape nicht darstellen; das Betätigen der Escape-Taste (<ESC>) bringt nicht den erhofften Effekt – DOS nutzt die Taste zum Abbrechen von Befehlen. Der Echo-Befehl scheidet also aus. Mit TYPE lassen sich Dateien auf den Bildschirm ausgeben, aber diese müssen erst mit einem Texteditor erstellt werden, und dann

Tafel 1 Überblick über Befehlsfolgen für ANSI.SYS

```
ESC [ x; y H      Setzen des Cursors an Zeile x, Spalte y
ESC [ x; y f      gleiche Wirkung
ESC [ x A         Bewegen des Cursors um x Zeilen nach oben
ESC [ x B         Bewegen des Cursors um x Zeilen nach unten
ESC [ x C         Bewegen des Cursors um x Spalten vor
ESC [ x D         Bewegen des Cursors um x Spalten zurück
ESC [ 6 n         Aktuelle Cursorposirion abfragen (nicht mit
                  Prompt verwenden)
ESC [ s           Sichern der aktuellen Cursorposition
ESC [ x; y R      Anzeigen der aktuellen Cursorposition
ESC [ u           Setzen des Cursors an die Stelle des
                  letzten Sicherns
ESC [ 2 J         Bildschirm löschen, Cursor home
ESC [ k           Bildschirm ab Cursorposition löschen
ESC [ = x h       Bildschirmmodus setzen
                  x     Wirkung
                  0     40x25 Zeichen schwarz/weiß
                  1     40x25 Zeichen Farbe
                  2     80x25 Zeichen schwarz/weiß
                  3     80x25 Zeichen Farbe
                  4     Grafik: 320x200 Punkte Farbe
                  5     Grafik: 329x200 Punkte schwarz/weiß
                  6     Grafik: 640x200 Punkte schwarz/weiß
ESC [ = x l       Bildschirmmodus zurücksetzen
ESC [ x; ...x m   Zeichenattribute setzen
                  x     Wirkung
                  0     zurücksetzen, weiß auf schwarz
                  1     höhere Helligkeitsstufe
                  2     niedrigere Helligkeitsstufe
                  3     Kursivschrift
                  4     unterstrichen (nur monochrom)
                  5     blinkend
                  6     schnelles Blinken
                  7     invers
                  8     löschen
                  48    hochgestelltes Zeichen
                  49    tiefgestelltes Zeichen
                  Vordergrund    Hintergrund
                  30             40             schwarz
                  31             41             rot
                  32             42             grün
                  33             43             gelb
                  34             44             blau
                  35             45             violett
                  36             46             kobaltblau
                  37             47             weiß
ESC [ x; ... p    Tastatur neu belegen
                  x gibt den Tastencode an, es folgt die
                  neue Belegung
```

gibt es die gleichen Probleme mit nichtdruckbaren Zeichen, wie Escape. Die Überschrift verrät es: Der Prompt-Befehl kann hier Abhilfe leisten. Seine eigentliche Aufgabe besteht bekanntlich in der Einstellung der DOS-Bereitschaftsanzeige; der Nebeneffekt der Ausgabe auf den Bildschirm wird für die Escape-Sequenzen genutzt. Vorteilhaft vor allem, daß das Escape-Zeichen eine eigene Darstellung erhält, nämlich $e (Dollarzeichen und kleines e). Wie Sie Tafel 1 entnehmen können, wird mit ESC [2 J der Bildschirm gelöscht. Die Leerzeichen werden nur zur besseren Lesbarkeit eingefügt und dürfen sich nicht im Kommando befinden, das folgendermaßen aussieht (die eckige Klammer erreichen Sie, wie in Teil 1 beschrieben, mit <ALT> 91):

prompt$e[2J

Wenn Sie alles richtig gemacht haben, ist der Bildschirm jetzt gelöscht und der Cursor steht links oben in der Ecke. Auffallen wird, daß kein Prompt mehr zu sehen ist – ihn haben wir mit dieser Anweisung außer Kraft gesetzt. Um Laufwerk und Pfad wieder angezeigt zu bekommen, muß

prompt$p$g

eingegeben werden. Vorläufig reicht aber auch einfach **prompt**, denn auf den Pfad können wir bei den folgenden Übungen verzichten. Hat der Befehl nicht die erhoffte Aktion ausgeführt, so überprüfen Sie bitte folgende Fehlerquellen: Steht der Treiber ANSI.SYS in der Konfigurationsdatei CONFIG.SYS (wie oben beschrieben)? Und haben sie weitere Programme geladen, die ihrerseits die Bildschirmausgabe kontrollieren?

Zumeist ist es der Norton-Commander (den wir uns in einer weiteren Kurs-Folge noch genauer ansehen werden), es können aber auch andere residente Programme wie Side-Kick oder Superkey daran Schuld sein. Entfernen Sie also zunächst einmal den Norton-Commander (Funktionstaste F10) und probieren Sie es noch einmal.

Nun ist das Löschen des Bildschirms keine allzu hilfreiche Anwendung – dafür gibt es ja den CLS-Befehl. Deshalb wollen wir als nächstes einmal einen anderen Bildschirmmodus einstellen, nämlich mit nur 40 Zeichen auf einer Zeile, die man dann auch ohne Brille ganz gut lesen kann. Dazu haben Sie

prompt $e[=0h

zu schreiben. Auch hier muß der Prompt wieder hergestellt werden, indem prompt eingegeben wird. Inzwischen müßten Sie in der Lage sein, aus Tafel 1 festzustellen, wie in den normalen Bildschirmmodus zurückgeschaltet werden kann. Normal sind 25 Zeilen zu 80 Zeichen (80 × 25). Da es in DOS häufig mehrere Wege zum Ziel gibt, können Sie das gleiche aber auch mit

mode 80

erreichen – in einer späteren Folge wird gezeigt, was es damit auf sich hat. Jetzt dürfte es keine Schwierigkeiten mehr bereiten, beliebige andere Einstellungen vorzunehmen, beispielsweise

prompt $e[7m

zur Einstellung einer inversen Darstellung. Ihnen wird auffallen, daß nicht sofort der ganze Bildschirm invers ist, sondern nur die Ausgaben ab aktueller Cursorposition.

Weitere Möglichkeiten ergeben sich, wenn Sie über einen Farbbildschirm verfügen. Das Auge freut sich sicherlich über gelbe Schrift auf blauem Grund, die mit

prompt $e[33;44m

eingestellt wird. Der Befehl kann gefahrlos auch auf einen Schwarzweißbildschirm angewendet werden. Er erzeugt Graustufen und sieht – je nach Bildschirmkarte – ganz passabel aus. Eine Einstellung ganz besonderer Art wird von

prompt $e[31;41m

vorgenommen. Probieren Sie es doch einmal aus. Die anschließende Eingabe müssen Sie blind vornehmen oder die Reset-Taste betätigen. Warum? Rote Schrift auf rotem Grund...

ANSI.SYS ist aber nicht nur für den Bildschirm verantwortlich, sondern auch für die Tastatur; Bildschirm und Tastatur werden zusammen Konsole genannt, und ANSI.SYS ist ein Konsol-Treiber.

Was läßt sich an einer Tastatur ändern? Nun, die Belegung der Tasten. Warum? Erinnern sie sich der Suche des Backslashs (\), der auf vielen Tastaturen nur schwer zu erreichen ist. Oder aber denken Sie an die Möglichkeit, eine ganze Zeichenkette (etwa einen Programmnamen) auf nur eine Taste zu legen. Dazu müssen Sie den Code wissen, den diese Taste abgibt. Alle Tasten, die ein druckbares Zeichen darstellen, können Sie unter ihrem ASCII-Code erreichen. Sie benötigen dazu eine ASCII-Tabelle, die in kaum einer Dokumentation fehlt. Dort werden Sie das kleine **u** als eine dezimale 117 und das kleine **x** als 120 finden. Als erstes fällt einem sicherlich ein Beispiel aus dem Leben ein; um also ein **x** für ein **u** zu erhalten, brauchen Sie nur noch

prompt $e[117;120p

einzugeben (siehe auch Tafel 1). Wie Sie sich selbst überzeugen, können trifft die Einstellung jetzt nur für das kleine u zu, für den Großbuchstaben ist die gleiche Prozedur mit seinem Tastencode nötig. Wenn Sie dieses Beispiel nicht überzeugt, so möchte ich Ihnen auch eine sinnvolle Anwendung vorstellen. Wir wenden uns deshalb dem Backslash zu und legen ihn auf das Ausrufezeichen, das uns damit verlorengeht:

prompt $e[33;92p

Beachten Sie bitte, daß vor dem letzten Buchstaben (in diesem Falle das **p** zur Kennzeichnung der Umbelegung) kein Semikolon steht. Da kaum ein Zeichen entbehrlich ist, wäre es viel interessanter, die Funktionstasten zu belegen. Diese Steuertasten besitzen einen sogenannten erweiterten Tastaturcode. Er beginnt stets mit einer Null und es folgt ein weiteres Zeichen (SCAN-Code). Gleiches gilt auch für die Buchstaben in Verbindung mit der ALT- oder Control-(Strg-)Taste. Für eine vollständige Tabelle fehlt uns der Platz – Ihnen bleibt nur der Blick in die Systemliteratur. Trotzdem als kleine Hilfe die wichtigsten Belegungen:

<F1> bis <F10>	Code: 0;59 bis 0;68
<F1> bis <F10> mit <SHIFT>	Code: 0;84 bis 0;93
<F1> bis <F10> mit <CONTROL>	Code: 0;94 bis 0;103
<F1> bis <F10> mit <ALT>	Code: 0;104 bis 0;113
<ALT> <Q>	Code 0;16
<ALT> <W>	Code 0;17
<ALT> <D>	Code 0;32
<ALT> <F>	Code 0;33

Sie sehen, daß die Codes nicht aufsteigend nach dem Alphabet, sondern nach der Anordnung auf der Tastatur vergeben worden sind. Die Abtrennung mit Semikolon wurde gewählt, da sie auch in den Escape-Sequenzen so gefordert ist. Damit Sie einen Vergleich haben, hier der Befehl um den Backslash auf die F1-Taste zu legen:

prompt $e[0;59;92p

Damit verliert die F1-Taste natürlich die in Teil 1 beschriebene Möglichkeit, einzelne Zeichen aus dem Tastaturpuffer zu holen. Und natürlich wird auch deutlich, daß sich solch eine Einstellung nicht mit dem Norton-Commander verträgt, der seine eigene Belegung der F1-Taste durchsetzen würde. Etwas sinnvoller lassen sich die Funktionstasten mit einem ganzen Kommando belegen, zum Beispiel mit dem DIR-Kommando, wiederum auf die Taste F1:

prompt $e[0;59;"dir";13p

Die dezimale 13 hinter dem Kommando ist ein Wagenrücklauf (Carriage Return) und bewirkt, wie das Betätigen der Enter-Taste, die sofortige Ausführung des Kommandos. Genausogut ist es auch möglich, auf die Eingabe weiterer Parameter von Hand zu warten. Das Kommando

prompt $e[0;60;"type"p

legt auf F2 den Befehl TYPE und wartet beim Betätigen der Taste auf Eingabe eines Dateinamens. Das Belegen der Tasten mit Kommandos beschränkt sich nicht auf die Funktionstasten. Dazu nutzt man die Mehrfachbelegung der Tasten, also <CONTROL> <Buchstabe> oder <ALT> <Buchstabe>. So stehen ausreichend Tasten für eigene Ideen zur Verfügung. Versuchen Sie es mal mit dem Kommando FORMAT A: auf die Tasten <ALT> <F>:

prompt $e[0;33;"format a:";13p

oder mit dem Aufruf von dBase auf <ALT> <D> mit

prompt $e[0;32;"dbase";13p

Aufmerksame Leser werden sicher auf die Idee kommen, nicht nur Systemkommandos, sondern auch sonst benötigte Zeichenketten fest auf Tasten zu legen, zum Beispiel „BEGIN" für Pascal-Programmierer, „LIST" für dBase oder „Mit freundlichen Grüßen" für eine beliebige Textverarbeitung. Hier werden Sie enttäuscht. Der ANSI-Treiber tut seine Arbeit nur auf der Ebene des Betriebssystems, innerhalb von Programmen stehen Ihnen die Belegungen nicht zur Verfügung, da die meisten Programme eigene Ein-/Ausgaberoutinen benutzen (Ausnahmen bestätigen wie immer nur die Regel).

Es gibt allerdings Hilfsprogramme, die sogenannte Tastaturmacros, also die Belegung von Tasten mit Zeichenketten in sehr komfortabler Form gestatten. Dazu gehören Super-Key und Keyworks. Und noch ein Hinweis: Es steht nicht unbegrenzt Platz für die Umbelegung von Tasten zur Verfügung. In DOS 3.x werden Sie im Schnitt auf 30, höchstens 64 Redefinitionen kommen. Aber das sollte ja ausreichend sein. Ist der dafür vorgesehene Puffer voll, werden die danach folgenden Angaben ohne Fehlermeldung ignoriert.

Haben Sie ein paar sinnvolle Belegungen gefunden, so schreiben Sie diese am besten gleich in die AUTOEXEC.BAT, um sie nach jedem Systemstart wieder zur Verfügung zu

haben, denn selbstverständlich gehen sie verloren, wenn der Rechner abgeschaltet wird.
Zu guter Letzt sei noch einmal daran erinnert, daß die Ausgabe von Steuersequenzen mit der eigentlichen Aufgabe des Prompt-Befehls nichts zu tun hat; er wird sozusagen zweckentfremdet benutzt. Der Vorteil liegt darin, daß wir nicht extra eine Programmiersprache bemühen müssen, sondern auf DOS-Kommandoebene arbeiten können. In einer Programmiersprache können die Steuerzeichen einfach mit dem Ausgabebefehl auf den Bildschirm ausgegeben werden. Die einzige Schwierigkeit besteht darin, daß der Programmierer wissen muß, wie die nichtdruckbaren Zeichen (vor allem ESCAPE) ausgegeben werden. Die Belegung der F1-Taste mit dem DIR-Befehl sei hier als Beispiel in Pascal und C gezeigt:

```
write(chr(27),'[0;59;"dir";13p');          {Pascal}
printf("%c[0;59;\"dir\";13p",27);  /*C*/
```

Achten Sie aber darauf, daß der Compiler die Programmierschnittstelle von DOS (BDOS) verwendet und nicht (an ANSI.SYS vorbei) auf das ROM-BIOS zugreift. Für Turbo-Pascal heißt das, auf die Unit CRT zu verzichten.

Stapelverarbeitung

Der Ihnen vielleicht etwas seltsam anmutende Begriff Stapelverarbeitung stammt noch aus Zeiten, als man wahrhaftig Lochkartenstapel in seinem Rechenzentrum abgab, um dafür (Tage später) eine Druckliste zu erhalten. Damit sich das aber nicht allzu profan anhört, sagte man auch Job dazu. Die PC-Nutzer geben sich modern und sagen Batch, was auch nichts anderes heißt als Stapel oder Stoß (Papier). Darunter wird eine Folge von Kommandos auf DOS-Ebene verstanden, die ohne Zutun des Nutzers als Block abgearbeitet werden. Ziel ist es, stets wiederkehrende Arbeiten zu automatisieren. Dazu können alle bisher besprochenen Befehle (interne und externe) sowie beliebige Programme verwendet werden. Zusätzlich stellt DOS noch Steuerstrukturen zur Verfügung, die die eigentliche Kommandosprache ausmachen, und um die soll es in diesem Abschnitt gehen. Vom Umfang her ist die DOS-Kommandosprache recht klein (verglichen mit denen anderer Betriebssysteme), aber sie kann sehr mannigfaltig eingesetzt werden und bildet so ein mächtiges Werkzeug.
Auch ohne weiteres Wissen können wir schon Batch-Dateien schreiben. Und eine ganz besondere haben wir schon kennengelernt: AUTOEXEC.BAT. Als nächstes soll eine Batch-Datei angelegt werden, die uns die verfügbaren externen Befehle auf der Festplatte anzeigt. Dazu brauchen wir nun aber doch einen Texteditor – der COPY-Befehl ist für größere Arbeiten nicht geeignet. Schreiben Sie also mit dem Editor Ihrer Wahl (Wordstar, Norton-Editor, Turbo-Editor) die neue Datei EXDIR.BAT mit dem Inhalt:

```
dir \dos\*.com /w
```

Anschließend können Sie die Datei mit ihrem Namen – also EXDIR – aufrufen, und es werden alle COM-Dateien aus dem Verzeichnis DOS (und dort stehen gerade die externen DOS-Befehle) in Kurzform angezeigt. Bei nur einem Befehl in der Kommandodatei ist die Arbeitserleichterung nicht sonderlich groß; wie weitere Befehle verwendet werden können, war schon bei der AUTOEXEC zu sehen. Was jedoch fehlt, sind Steuerstrukturen für die Wiederholung von Kommandos und zum Prüfen von Bedingungen. Für letzteres gibt es in DOS den **IF**-Ausdruck. Es können drei verschiedene Arten von Bedingungen getestet werden: die Existenz von Dateien (mit EXIST), die Rückgabe von Fehlernummern durch Programme (ERRORLEVEL) und die Gleichheit von Zeichenketten (==), was besonders für Parameter und Dateinamen verwendet werden kann. Durch den Einsatz von **NOT** können die Aussagen auch jeweils negiert werden. Wenn Sie die Datei CONFIG.SYS listen wollen, aber keine Fehlerausschrift, wenn diese nicht vorhanden ist, dann geht das mit

```
if exist config.sys type config.sys
```

Diese Befehlsfolge kann auch interaktiv eingegeben werden. Im folgenden wird nicht weiter darauf hingewiesen, diese Anweisungen in eine BAT-Datei zu schreiben und zu starten.
Genauso kann auch eine Aktion ausgeführt werden, wenn die Datei nicht vorhanden ist. Oder gleich in beiden Fällen:

```
if exist config.sys type config.sys
if not exist config.sys
echo CONFIG.SYS ist nicht vorhanden
```

Die Unterdrückung der Anzeige der Befehle erfolgt mit **echo off**. Dieser Befehl steht in der Regel in der ersten Zeile einer Stapeldatei – am Ende wird das Echo automatisch wieder eingeschaltet. Noch schöner wäre dieser Batch, wenn er nicht nur auf eine Datei anwendbar wäre, sondern auf beliebige. Dazu geben wir den Dateinamen als Parameter beim Aufruf an. In der Batchdatei steht dafür ein Platzhalter. Die Parameter (bis zu neun Stück) werden mit %1 bis %9 bezeichnet. Der gleiche Batch wie oben für beliebige Dateien sieht so aus:

```
if exist %1 type %1
if not exist %1 echo %1 ist nicht vorhanden
```

Der ECHO-Befehl wird – wie bereits gezeigt – entweder dazu genutzt, die Ausgaben der Befehle zu unterdrücken (ON/OFF) oder aber um Meldungen auf den Bildschirm zu bringen. Der Aufruf der angenommenen Datei TEST.BAT erfolgt dann mit

```
test dateiname
```

Sie listet als **test autoexec.bat** entweder die Datei oder schreibt, daß sie nicht vorhanden ist. Als nächstes soll verhindert werden, daß man auch die CONFIG.SYS damit ansehen kann. Alle anderen Dateien ja, aber die Konfigurationsdatei soll nicht zugänglich sein. Dazu sollte es genügen, das Kommando

```
if %1 == config.sys echo Diese Datei ist tabu
```

voranzustellen (beachten Sie bitte, daß es sich um zwei Gleichheitszeichen handelt). Beim Aufruf von **test config.sys** werden Sie feststellen, daß die Ausschrift wirklich kommt, die Datei aber anschließend trotzdem gezeigt wird (sofern sie vorhanden ist). Ursache dafür ist, daß alle Befehle nacheinander durchlaufen werden. Um dem zu begegnen, kann mit **GOTO** verzweigt werden, indem eine Sprungmarke angegeben wird. Sie kann beliebig lang sein, aber nur 8 Zeichen werden zur Auswertung herangezogen. Die Marke selbst wird von einem Doppelpunkt eingeleitet. Wird sie nicht gefunden, so bricht DOS mit einer Fehlermeldung ab. Das Beispiel von oben sieht dann richtig so aus:

```
echo off
if %1 == config.sys goto tabu
if exist %1 type %1
if not exist %1 echo %1 ist nicht vorhanden
exit
:tabu
echo Diese Datei ist tabu.
```

Beim Dateinamen (wie bei allen Parametern) muß Groß- und Kleinschreibung unterschieden werden.
Als Beispiel für die Übergabe mehrerer Parameter schreiben wir uns jetzt eine Befehlsdatei, die mehrere Dateien listet:

```
if exist %1 type %1
if exist %2 type %2
if exist %3 type %3
```

Beim Aufruf mit

```
test config.sys autoexec.bat brief.txt
```

werden alle drei Dateien nacheinander angezeigt. Damit können sie umgehen, daß beim TYPE-Kommando keine Joker erlaubt sind. Trick am Rande: Zum Listen mehrerer Dateien kann auch

```
copy dateien con
```

verwendet werden, beispielsweise

```
copy *.bat con
```

zum Listen aller BAT-Dateien. Wenn Ihnen das nicht einleuchtet: Einmal aussetzen und zwei Folgen zurück zum COPY-Befehl.
Wie bereits erwähnt, können die Parameter %1 bis %9 verwendet werden (%0 enthält den Namen der BAT-Datei selbst). Sollten Sie wirklich einmal nicht damit auskommen, so kann der **SHIFT**-Befehl zur Anwendung kommen. Er verschiebt die Parameterkennungen nach links. Sie erreichen unter %1 dann den zweiten Parameter und unter %9 den zehnten. Durch mehrmaliges Anwenden können so alle Parameter zugänglich gemacht werden. Für die meisten Anwendungen wird das aber kaum nötig sein.
Unbefriedigend beim Anzeigen der drei Dateien ist, daß stets genau drei Parameter angegeben werden müssen. Besser wäre es, wenn bei einem weggelassenen Parameter die Aktion nicht ausgeführt wird. Dazu wird die Parameterzeichenkette daraufhin geprüft, ob sie leer ist, und zwar mit **if %1* == * ...** Eine leere Zeichenkette im eigentlichen Sinne ist nicht vorhanden. Deshalb wird ein beliebiges Zeichen angehängt (hier ein Stern) und verglichen, ob es sich nur um einen Stern handelt. Zur besseren Lesbarkeit ist aber auch **if "%1" == "" ...** möglich. Der Batch zum Anzeigen von 1 bis 3 Dateien hat dann folgendes Aussehen:

```
if exist %1 type %1
if "%2" == "" goto ende
if exist %2 type %2
if "%3" == "" goto ende
if exist %3 type %3
:ende
```

Wenn Sie genau aufgepaßt haben, werden Sie feststellen, daß jetzt doppelt geprüft wird. Denn ein nicht übergebener Parameter wird als leere Zeichenkette interpretiert und deshalb als Datei nicht gefunden, was auch keine Fehlermeldung auslöst. In vielen Fällen ist es aber wichtig, auf die Existenz eines Parameters zu prüfen. So würde die Anweisung **format %1:** beim Fehlen des Laufwerksparameters **format :**ergeben. Wird die Existenz des Para-

Befehl	Funktion
CALL datei [parameter].	Aufruf einer weiteren Batch-Datei (ab DOS 3.30)
GOTO marke :marke	Verzweigen zu einer Sprungmarke
ECHO [ON | OFF | meldung]	Ausgabe von Meldungen auf den Bildschirm
EXIT	Vorzeitiges Beenden einer Batch-Datei
FOR %%variable IN (parameter) DO kommando	Wiederholte Abarbeitung von Kommandos, solange die Bedingungen erfüllt sind (Bei direkter Abarbeitung nur ein Prozentzeichen)
IF [NOT] bedingung kommando	Bedingte Ausführung von Kommandos
bedingung: ERRORLEVEL n	Rückgabewert größer oder gleich n
EXIST dateiname	Vorhandensein einer Datei
string1 == string2	Gleichheit von Zeichenketten
PAUSE [meldung]	Warten auf Tastendruck
REM kommentar	Kommentare in Batch-Dateien (keine Wirkung)
SHIFT	Verschieben der Parameter nach links

meters getestet, ergibt sich die Möglichkeit, einen Standard (z. B. a:) anzunehmen.
Nicht unterstützt von der DOS-Kommandosprache werden Tastatureingaben, die dem Nutzer die Auswahl aus einem Menü oder die Beantwortung einer Abfrage (etwa: „Wirklich Formatieren (J/N)") gestatten. Deshalb wird zu einem Trick gegriffen, den wir – obwohl häufig in Publikationen gezeigt – wegen seiner Nützlichkeit kurz vorstellen möchten. Diesem liegt die Tatsache zugrunde, daß mit **ERRORLEVEL** der Programmrückkehrcode im Prozessorregister AL ausgewertet wird. Beispielsweise gibt Format (wie den DOS-Unterlagen zu entnehmen ist) eine 0 für fehlerfreie Beendigung der Arbeit, eine 3 für einen Abbruch mit <CTRL> <C> bzw. <Strg> <C> und eine 4 für einen Diskettenfehler zurück. Das läßt sich mit folgendem Batch-Programm ausprobieren:

```
format a:
if errorlevel 4 goto fehler
if errorlevel 3 goto abbruch
if errorlevel 0 goto okay
```

usw.
Um Tastatureingaben auszuwerten, wird nur ein kleines Programm benötigt, daß den Tastendruck abfragt und ASCII-Dode übergibt. Am einfachsten wird folgendes Assemblerprogramm mit einem Editor erfaßt und durch den Debugger Debug (befindet sich unter den externen DOS-Kommandos) geschickt. Das ist auch für Nutzer mit wenig Erfahrung zu schaffen. Hier das Programm TASTE.ASM:

```
a
mov ah,0
int 16h
mov ah,4ch
int 21h
          (Leerzeile)
r cx 8
n taste.com
w
q
```

Übersetzt wird es durch
debug < test.asm
und das fertige Programm TASTE.COM kann dazu verwendet werden, den Tastencode als ERRORLEVEL zurückzugeben. Es gibt sicher elegantere Lösungen, aber keine kürzere. Sie brauchen nur noch in der ASCII-Tabelle den Zeichencode nachzusehen. Beachten Sie dabei, daß Tasten, die einen erweiterten Code abgeben (Funktionstasten, Kombinationen mit <ALT> oder <CTRL>) keine Verwendung finden können. Das eigentliche Ziel bei der Einführung des ERRORLEVEL war, auf eine fehlerhafte Beendigung eines Programms reagieren zu können. Als Beispiel wird häufig das Assemblieren von Programmen gezeigt und je nachdem, ob das fehlerfrei geschieht oder nicht, wird zum Linken oder zum erneuten Editieren verzweigt. Wir benutzen den ERRORLEVEL-Ausdruck also wieder einmal zweckentfremdet (wie schon den Prompt-Befehl). Dazu wird ein kleines Menü erzeugt, aus dem der Anwender ein Programm aussuchen kann, das gestartet werden soll (in diesem Falle Wordstar, dBase und Multiplan). Die Auswahl erfolgt durch Eingabe des Anfangsbuchstabens (Programm Taste). Und so könnte eine kleine Batch-Datei aussehen:

```
echo off
echo Aufruf von Programmen
echo.
echo w – Wordstar
echo m – Multiplan
echo d – dBase
echo.
echo Bitte Auswahl:
taste
if errorlevel 119 goto ws
if errorlevel 109 goto mp
if errorlevel 100 goto db
exit
:ws
c:\text\ws
exit
:mp
c:\calc\mp
exit
:db
c:\data\dbase
```

Aus Platzgründen wurde auf jegliche Verschönerungsmaßnahmen verzichtet. Statt des einfachen Programmstarts sollten noch die Standarddirectories eingestellt, Suchpfade festgelegt und eine nette Ausschrift gegeben werden.
Mit **echo.** wird einfach eine Leerzeile erzeugt. Und nun die Auswertung: **if errorlevel 119** testet, ob ein Code größer (!) oder gleich 119 übergeben wurde. Die 119 ist gerade ein kleines w in der ASCII-Tabelle. Dieses w bewirkt dann den Start von Wordstar. Anschließend wird auf m (109) und d (100) geprüft. Wichtig ist die Aussage, daß stets auf größer oder gleich untersucht wird. Das hat den Hintergrund, daß beim Test auf Fehler nicht alle möglichen Codes abgefragt werden können. Ein fehlerfreies Programm gibt 0 zurück, und mit aufsteigenden Nummern werden die Fehler schwerwiegender. Auf diese Weise müssen zuerst die schweren Fehler bearbeitet werden, z. B.

```
if errorlevel 8 goto abbruch
if errorlevel 4 goto warnung
if errorlevel 0 goto okay
```

So wird bei allen schlimmeren Fehlern als 8 abgebrochen, bei Fehler zwischen 4 und 8 gewarnt usw.
Für unser obiges Beispiel heißt das aber, daß bei allen Tasten mit einem Code größer als 119 zu Wordstar verzweigt wird. Also zum Beispiel, wenn Sie ein z eingeben. Und bei einem p gehen Sie zu Multiplan – ein kleiner Nachteil dieses kurzen Programms. Ein oft begangener Fehler bei dieser Art der Batch-Programmierung ist, daß die Prüfung mit ERRORLEVEL nicht in absteigender Reihenfolge geschieht. Dann landen Sie stets im ersten Menüpunkt.
Wenn Ihnen der Umgang mit dem Programm TASTE zu schwierig erscheint, so können sie auch ASK aus den Norton-Utilities benutzen. Das oben gezeigte Programm sieht dann (leicht gekürzt) so aus:

```
echo w – Wordstar
echo m – Multiplan
echo d – dBase
ask "Bitte Auswahl", dmw
if errorlevel 3 goto ws
if errorlevel 2 goto mp
if errorlevel 1 goto db
```

usw.
Dem Programm ASK werden neben einer Ausschrift die Anfangsbuchstaben der möglichen Antworten übergeben, und man erhält die Nummer des gedrückten Buchstabens zurück (hier nicht der ASCII-Code). Weiterhin muß aber in absteigender Reihenfolge getestet werden.

Literatur

/1/ MS-DOS-Referenzkarte. Mikroprozessortechnik, Berlin 2 (1988) 12, 3. US
/2/ Smode, D.: Das große MS-DOS-Profi-Arbeitsbuch. München: Franzis-Verlag GmbH 1987

In der nächsten Folge bleiben wir noch ein wenig bei der Stapelverarbeitung, und Sie erfahren dann, was es mit Filtern und Pipes in DOS auf sich hat und daß Umleitungen nützlich sein können.

Einführung in MS-DOS

Teil 4

Wolfram Schulze, Uwe Schulze, Berlin

Was spielt sich in der Umgebung von Programmen ab? Umleitungen zum Nutzen des Programmierers? Filter für DOS? Antworten in diesem Kurs. Doch zunächst noch einmal zur Batch-Programmierung.

Wenn in einem Batch-Programm nur auf einen Tastendruck gewartet werden soll, ohne daß weiter ausgewertet wird, so verwendet man am einfachsten den Befehl **PAUSE**:
pause Bitte Diskette in a: einlegen
Die Meldung wird als Kommentar interpretiert und auf den Bildschirm ausgegeben, sofern ECHO ON geschaltet ist. DOS fordert mit **Weiter → eine Taste betaetigen** zur Weiterarbeit auf.
Mit **GOTO** lassen sich zwar Wiederholschleifen anlegen, aber die Festlegung von Abbruchbedingungen und die Übergabe anderer Parameter für jeden Umlauf gestalten sich umständlich. Deshalb stellt DOS die **FOR-DO**-Schleife zur Verfügung. Um beispielsweise alle Pascal-Quelltexte (Erweiterung .PAS) mit dem Turbo-Pascal-Kommandozeilencompiler TPC zu übersetzen, ist folgende Anweisung nötig.
for %%f in (*.pas) do tpc %%f
Solange noch .PAS-Dateien gefunden werden, wird der Aufruf **TPC dateiname** ausgeführt. Zur Belegung mit dem Dateinamen wird die Variable %%f benutzt. Sie können aber auch jeden anderen Buchstaben mit zwei vorangestellten Prozentzeichen verwenden (im Gegensatz zu Parametern, die aus einem Prozentzeichen und einer Ziffer bestehen). Wird ein FOR-Befehl aber interaktiv eingegeben (also direkt auf der DOS-Kommandoebene und nicht in einer BAT-Datei), so ist nur ein Prozentzeichen zu schreiben.
Des weiteren kann der FOR-Befehl dazu verwendet werden, Aufgaben auszuführen, die allein mit den Jokern nicht gelöst werden können. Zum Beispiel das Anzeigen aller Dateien mit den Erweiterungen .TXT und .DOC. in einem Befehl. Hier ist er:
for %%f in (*.txt,*.doc) do type %%f
Oder das Kopieren aller Programmdateien (also Erweiterungen COM, EXE und BAT) nach A:
for %%f in (*.com,*.exe,*.bat) do copy %%f a:%%f
Das Schachteln von Kommandos ist ebenfalls kein Problem. Wenn nur die Dateien auf die Sicherungsdiskette in A: kopiert werden sollen, die dort noch nicht vorhanden sind, so geschieht das mit
for %%f in (*.*) do if not exist a:%%f copy %%f a:
In großen Stapeldateien ist es zweckmäßig, durch Kommentare spätere Änderungen zu erleichtern. Dafür ist REM (Remark) vorgesehen. Kommentarzeilen können an jeder beliebigen Stelle eingefügt werden; sie beeinflussen den Programmverlauf nicht und erscheinen auch nicht auf dem Bildschirm. Beispiel:
rem Batch nach Anregung aus der MP
In Stapeldateien können neben den DOS-Befehlen auch beliebige Programme, also auch andere Stapeldateien, aufgerufen werden. Der Ruf anderer Batch-Dateien hat aber einen unerwünschten Nebeneffekt: Es erfolgt keine Rückkehr in das rufende Programm. In der gewünschten Weise kann ein Aufruf weiterer BAT-Dateien also nur am Ende (als letzter Befehl) erfolgen. Das hängt mit der internen Organisation der Abarbeitung von Stapeldateien zusammen. Ab DOS-Version 3.30 steht ein neuer Befehl zum Aufruf von Stapeldateien als Unterprogramm zur Verfügung: **CALL**. Rufen Sie also aus einer BAT-Datei die Datei TEST.BAT mit
call test
auf. Und eine weitere Verbesserung gibt es ab Version 3.30. Mit einem vorangestellten @ kann die Ausgabe einer Kommandozeile auf den Bildschirm unterdrückt werden. Selbst wenn Sie als erstes echo off stellen erscheint eben dieser Befehl noch auf dem Bildschirm. Schreiben Sie deshalb
@echo off
Auch vor andere Befehle kann ein @ gesetzt werden, das dann jeweils nur für diese Zeile gilt.
Benötigen Sie vor Version 3.30 den Call-Befehl, so schreiben Sie statt dessen
command /c test
Damit wird der Kommandointerpreter geladen und die Batch-Datei als Parameter übergeben – allerdings ist mehr Speicherplatz nötig.
Batch-Dateien können stets mit <CTRL> <C> (<Strg> <C>) unterbrochen werden, auch wenn BREAK OFF eingestellt ist. Dann wird aber nicht der aktuelle Befehl erreicht, sondern erst vor dem nächsten Befehl die Ausschrift
Stapeldatei abbrechen (J/N)
ausgegeben. Und weil wir gerade beim Abbrechen sind: Gerade in Batch-Dateien kann es schnell zu Laufwerksfehlern kommen, zumeist, weil keine (formatierte) Diskette im Laufwerk liegt. Man erhält die Ausschrift
Fehler beim Schreiben in Laufwerk A
Abbrechen, Wiederholen, Fehler
Die Eingabe des Anfangsbuchstabens genügt. Was unterscheidet diese Möglichkeiten? Mit A für Abbrechen wird der aktuelle Befehl beendet. In den meisten Fällen ist dieser Variante der Vorzug zu geben, weil sie keinen Schaden anrichtet. Interessante Möglichkeiten werden mit F für Fehler geboten. Hier wird die Fehlernummer an das rufende Programm (in diesem Falle die Batch-Datei) übergeben. Dort können sie wie oben beschrieben mit ERRORLEVEL getestet und entsprechende Maßnahmen eingeleitet werden. Viel ist damit im Falle eines Laufwerksfehlers nicht erreicht, aber es könnte zum Beispiel eine detaillierte Fehlermeldung gebracht werden. Wiederholen (w) kann zum Beispiel verwendet werden, wenn die Diskette mit einem Schreibschutz versehen war und nach dem Entfernen weiter gearbeitet werden soll. Aber Vorsicht: Gefährlich kann es sein, die Diskette zu wechseln und dann W zu drücken. In diesem Falle können im Speicher befindliche Informationen der alten Diskette (Directory) geschrieben und die Informationen der neuen Diskette zerstört werden.
Beachten Sie bitte, daß Batch-Dateien nicht den Namen bereits vorhandener Befehle (z. B. DIR.BAT) erhalten dürfen. Bei Eingabe eines Kommandos prüft DOS zuerst, ob es sich um einen internen (und damit reservierten) Befehl handelt. In diesem Falle wird dieser aufgerufen. Auch wenn der Name eines externen Befehls oder eines beliebigen Programms genutzt wird, kommt die BAT-Datei nicht zum Zuge. DOS sucht stets zuerst nach einer COM-Datei, dann nach EXE und erst ganz zum Schluß nach einer eventuellen BAT-Datei.
Da Sie nun schon erfahrene DOS-Programmierer sind, zum Schluß noch ein Trick zum Nachdenken. Wenn Sie einmal Zeit und Datum einer Datei ändern wollen, so ist das recht müßig. Sie müßte in einen Editor geladen und wieder gespeichert werden, damit DOS automatisch die neue Zeit setzt. Und was machen Sie mit einer Datei, die nicht aus Text besteht? Mit dem Befehl
copy /b datei +,,
setzen Sie Datum und Zeit einer oder mehrerer Dateien auf den aktuellen Stand. Der Schalter /b ist für eben diese Nicht-Textdateien zuständig, die sonst erheblich zusammengekürzt werden, weil das Dateiende falsch interpretiert wird. Probieren Sie doch mal. Ein bißchen mehr über Batch-Programmierung können Sie auch in MP 3/89, Seite 171, lesen.

Die Programmumgebung

Um gewisse Rahmenbedingungen für Programme festschreiben zu können, gehört zu jedem Programm ein kleiner Speicherbereich, in dem globale Parameter eingetragen sind. Er wird Programmumgebung (englisch Environment) genannt. Auch wenn noch kein Programm gestartet wurde, so ist zumindest der Befehlsinterpreter COMMAND.COM aktiv, und selbstverständlich besitzt auch er eine Programmumgebung. Einen Blick in das Environment gestattet der SET-Befehl. Schreiben Sie einfach
set
und es wird die aktuelle Programmumgebung angezeigt, die etwa so aussehen könnte:
COMSPEC=C:\COMMAND.COM
PATH=C:\DOS;C:\TOOLS
PROMPT=pg
Diese drei Befehle sind auch die einzigen, die ohne den SET-Befehl in die Programmumgebung eingetragen werden.
Vielleicht steht auch noch etwas mehr drin, je

nachdem, was Sie – wahrscheinlich in der AUTOEXEC.BAT – eingestellt haben. Sind vom Nutzer keine Eintragungen vorgenommen worden, so schreibt DOS zumindest diese beiden Zeilen ein:

PATH=
COMSPEC=C:\COMMAND.COM

Der Pfad ist also leer und als Befehlsinterpreter wird COMMAND.COM aus dem Hauptverzeichnis des Bootlaufwerks geholt. Es kann durchaus sinnvoll sein, diese Einstellung zu ändern, vor allem, wenn nicht von der Festplatte gebootet wurde. Der Kommandointerpreter COMMAND.COM besteht aus zwei Teilen, einem stets im Speicher verbleibenden (residenten) Teil und einem sogenannten transienten Teil, der von anderen Programmen überschrieben werden kann, da er während der Abarbeitung anderer Programme nicht benötigt wird. So kann Speicherplatz gespart werden. Nach dem Beenden eines Programms muß DOS stets diesen Teil des Kommandointerpreters nachladen, und zwar vom Startlaufwerk. Wurde von A: gestartet, so ist ein oftmaliger Diskettenwechsel nötig. Mit COMSPEC kann deshalb eingestellt werden, wo nach COMMAND.COM gesucht werden soll. Günstig ist es, die Festplatte oder aber eine RAM-Diskette anzugeben. Die Verwendung einer RAM-Disk bringt außerdem Geschwindigkeitsvorteile und kann deshalb auch empfohlen werden, wenn von der Platte gebootet wurde. Dazu ist in der AUTOEXEC.BAT der Kommandointerpreter auf die RAM-Disk zu kopieren und COMSPEC einzustellen, z. B.

copy command.com e:
set comspec=e:\command.com

Zusätzlich zu den Systemvariablen kann der Nutzer beliebige eigene Variablen belegen; wiederum mit dem SET-Befehl, beispielsweise

set tag=montag oder
set name=pit oder
set codewort=roger

Der Variablen *tag* wird die Zeichenkette *montag* zugeordnet. Die erfolgreiche Eintragung kann man sich anschließend mit

set

ansehen. Dabei werden Sie sehen, daß DOS die Variablennamen stets in Großbuchstaben umwandelt, für die Belegung aber Groß- und Kleinschreibung unterscheidet, wie bei der Eingabe festgelegt.

Jedes Programm erbt das Environment von dem Programm, von dem es gestartet wird, also im allgemeinen von COMMAND.COM. Beim Beenden wird es wieder freigegeben, und das übergeordnete Environment wird aktiviert. Das hat zur Folge, daß alle inzwischen vorgenommenen Eintragungen verloren gehen, eine Tatsache, die häufig zu Fehlern führt. Von dieser Regelung sind beispielsweise SET-Befehle in Batch-Dateien betroffen. Da eine BAT-Datei beim Start eine eigene Programmumgebung erhält, werden alle Eintragungen dort vorgenommen. Nach dem Beenden befinden Sie sich wieder in der Programmumgebung des Kommandointerpreters, wo diese Einträge nicht mehr existieren. Positiv gesehen brauchen in Batch-Dateien gesetzte Systemvariablen am Ende also nicht gelöscht zu werden.

Bestes Beispiel für das Verschwinden von Systemvariablen ist die Arbeit mit dem Norton-Commander. Nutzen Sie den SET-Befehl bei geladenem Norton-Commander, so gelangen die Eintragungen nie in das Environment von COMMAND.COM oder in das des Norton-Commanders. Sie existieren nur unmittelbar während der Abarbeitung des SET-Befehls. Das hängt damit zusammen, daß bei jeder Befehlsausführung eine weitere COMMAND.COM-Kopie geladen und mit eigenem Environment versehen wird. Also Vorsicht.

An einem kleinem Beispiel kann man sich das Vererbungskonzept des Environments verdeutlichen. Schreiben Sie in eine BAT-Datei

set test=gesetzt
set

Beim Start wird die Variable TEST korrekt angezeigt. Wenn Sie aber mit *set* noch einmal nachsehen, so ist die Variable nicht mehr vorhanden. Sie existierte nur während der Zeit der Abarbeitung des Programms in seiner Programmumgebung.

Um die Nutzung der so eingetragenen Informationen in einem Programm muß sich der Nutzer selbst kümmern, denn weder das Betriebssystem noch Anwenderprogramme kennen die Variablen TAG, NAME oder CODEWORT. In den meisten Programmiersprachen existieren Konstrukte zum Abfragen von Environment-Variablen (GETENV in C, ENVSTR ab Turbo-Pascal 5.0), leider gibt es aber keine legale Möglichkeit, das Environment zu verändern. Das hängt damit zusammen, daß durch den Vererbungsmechanismus diese Eintragungen zum Programmende sowieso verlorengehen.

Einfacher als in Programmiersprachen lassen sich Informationen aus der Programmumgebung in Batch-Dateien nutzen. Dort wird der Variablenname einfach in Prozentzeichen eingeschlossen. Der Test auf das gesetzte Codewort kann in einer Batch-Datei so aussehen:

if not %codewort% == roger echo Unberechtigter Zugriff

Oder die Nutzung des Tages mit

echo Heute ist %tag%

Sollen Variablen stets präsent sei, so können diese SET-Befehle in die AUTOEXEC.BAT eingetragen werden.

Ein Entfernen aus dem Environment erfolgt durch Belegung mit einer leeren Zeichenkette – zum Löschen des TAG-Eintrages also

set tag=

Den Erfolg kann man sich mit

set

ansehen. Einige Programme kennen auch vordefinierte Variablen, die der Nutzer zur Einstellung von Standards belegen kann. Bei Compilern stehen häufig die Variablen INCLUDE (für den Suchpfad der Include-Dateien) und LIB (für den Suchpfad der Bibliotheken) zur Verfügung. Durch Anweisungen wie

set include=c:\compiler\inc
set lib=c:\compiler\bibl

kann dem Compiler mitgeteilt werden, wo er nach den entsprechenden Dateien zu suchen hat, ohne das bei jedem Aufruf angeben zu müssen.

Ein gestartetes Programm erhält ein eigenes Environment, und zwar als Kopie vom startenden Programm. Wenn also dBase aufgerufen wird, so erhält es eine Kopie der Programmumgebung von COMMAND.COM. So gehen keine Informationen verloren. Arbeiten mehrere Programme gleichzeitig (Norton-Commander, SideKick etc.), so verwaltet jedes Programm sein eigenes Environment. Zu beachten ist dabei, daß seine Größe beschränkt ist. Standardmäßig ist es 160 Byte groß. Probieren Sie doch einmal aus, einige sehr lange Parameter einzutragen:

set x=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx...usw.

Beim Ansehen mit *set* werden Sie feststellen, daß der Rest abgeschnitten wurde. In neueren Betriebssystemversionen wird gar eine Fehlermeldung ausgegeben, wenn die Programmumgebung voll ist. Da dem Environment eine wachsende Bedeutung beigemessen wird, ist ab DOS-Version 3.10 die Größe einstellbar (bis maximal 32 KByte). Das Einstellen kann immer geschehen, wenn COMMAND.COM geladen wird (siehe Folge 1), zum Beispiel

command /e:512

zur Festlegung von 512 Byte Environment. Soll diese Einstellung stets beim Start Bestand haben, so haben Sie

shell=command.com /e:512 \p

in die CONFIG.SYS einzutragen. Ein etwas größeres Environment ist bei vielen neueren Programmen nötig; gehen Sie aber auch nicht zu großzügig mit dem Speicherplatz um, es ist zu bedenken, daß jedes nachfolgend gestartete Programm eine ebenso große Kopie erhält.

Mit dem Shell-Befehl ist es auch möglich, ein anderes Programm als COMMAND.COM zur Kommandooberfläche zu erklären, etwa ein selbstgeschriebenes. Es funktioniert auch, den Norton-Commander dort festzulegen, aber alle Systembefehle (DIR,...) stehen nicht mehr zur Verfügung, weil sie nur von Norton an COMMAND.COM weitergegeben werden, und verlassen kann man den Commander auch nicht, weil kein Kommandointerpreter „darunterliegt".

Umleitung von Ein- und Ausgabe

Bis jetzt wurde bei der Bedienung ganz selbstverständlich vorausgesetzt, daß alle Eingaben von der Tastatur kommen und alle Ausgaben auf dem Bildschirm erscheinen. Warum eigentlich? Es gibt doch noch andere Möglichkeiten: Drucker, serielle Schnittstellen und Dateien. DOS betrachtet Bildschirm und

```
SET                  -  Anzeigen des Environments

SET variable=string  -  Eintragung einer Variablen
                        in das Environment

SET variable=        -  Löschen einer Eintragung
```

Bild 1

Tastatur (zusammen Konsole) als Standard für Ein- und Ausgabe. In Beschreibungen findet man deshalb auch STDIN (für Standard Input) und STDOUT (für Standard Output). Um eine möglichst flexible Handhabung der Ein- und Ausgabe zu erreichen, wurden ab der Version 2.00 Methoden zur Umleitung (englisch Redirection) von Ein- und Ausgaben aufgenommen. So können mit dem Kleinerzeichen (<) die Herkunft der Eingaben und mit dem Größerzeichen (>) das Ziel der Ausgaben festgelegt werden. Man spricht auch von der Steuerung eines Datenstromes. In Teil 1 (in MP 4/90) sind im Zusammenhang mit dem COPY-Befehl die DOS-Geräte bereits vorgestellt worden:
PRN oder LPT1 für den Drucker, COM1 oder AUX für die serielle Schnittstelle sowie NUL als „Papierkorb". Um also das aktuelle Verzeichnis nicht auf den Bildschirm (wie gewohnt), sondern auf den Drucker auszugeben (Einschalten nicht vergessen!), schreiben Sie

dir > prn

Mit <CTRL> <P> (<Strg> <P>) geht's auch – trotzdem eine wertvolle Möglichkeit. PRN und LPT1 sowie COM1 und AUX können synonym verwendet werden.
Das Gerät NUL ist ein Scheingerät, das die umgeleiteten Daten ganz einfach verschwinden läßt. Nicht besonders wichtig, aber auch nicht sinnlos. Man kann sich damit von unliebsamen Ausschriften befreien; nützlich in langen und stets wieder verwendeten Batch-Dateien. Wenn Sie alle Dateien des Tageswerkes mit

copy *.* a: > nul

auf eine Sicherungsdiskette retten, so entfällt die Ausschrift

xxx Datei(en) kopiert

Eine Besonderheit von DOS ist, daß eine völlige Gleichbehandlung von Geräten (Konsole, Drucker) und Dateien erfolgt. An jeder Stelle, wo ein Gerät angegeben wird, kann auch eine Datei stehen. Sie können Ihr Inhaltsverzeichnis also auch in eine Datei schreiben. Versuchen Sie es mal mit

dir > dir.dat

Es wird eine Datei DIR.DAT angelegt, in der das steht, was sonst auf den Bildschirm kommt. Schauen Sie doch mal rein – mit einem Texteditor oder mit

type dir.dat

Auf diese Weise habe ich beispielsweise die Originalausschriften der Programme aufgefangen, um Sie im Kurs darzustellen. Für die Beschreibung von CHKDSK in Teil 2 (in MP 6/90) wurden die Ausschriften mit

chkdsk > text

in eine Datei geleitet und dann in den Kurs-Text kopiert.
Die Umleitung von Ein- und Ausgabe ist aber nicht auf DOS-Kommandos beschränkt; grundsätzlich gilt gleiches für beliebige Programme – vorausgesetzt ihre Ausgaben gehen über DOS. Sie werden aber feststellen, daß viele Programme eigene Routinen für die Ein- und Ausgabe verwenden. Angesichts Fenstertechnik und Mausbedienung: Tendenz steigend. Im Zweifelsfalle hilft nur probieren.
Die Umlenkung der Eingabe läßt sich nur selten sinnvoll anwenden. Eine der wenigen Anwendungen wurde in der letzten Folge beim Programm TASTE gezeigt.
Ist nicht beabsichtigt, bei der Ausgabeumleitung eine neue Datei anzulegen, so kann auch der Anfügeoperator (zwei Größerzeichen) eingesetzt werden. Beispiel zum Schreiben der Verzeichnisse aller Laufwerke in eine Datei:

dir a: > inhalt
dir b: >> inhalt
dir c: >> inhalt

Der erste Befehl schreibt die Ausgaben in eine Datei, die neu angelegt wird. Die weiteren Kommandos fügen ihre Ausgaben an diese Datei an, die damit immer länger wird. Die vorherigen Eintragungen gehen nicht verloren. Das Verketten von Dateien kann ja auch mit COPY und + vorgenommen werden, aber das Anfügen von Ausgaben eines Programms wird nur so erreicht. Die bis jetzt kennengelernten vielfältigen Möglichkeiten erlauben es auch, die gleiche Aktion mit verschiedenen Befehlen zu erreichen. So erfolgt die Ausgabe einer Datei auf den Drucker sowohl mit

copy datei prn als auch mit
type datei > prn

Ist der gleiche Effekt auch mit

copy datei con > prn

zu erreichen? Sie werden sehen, daß die Datei selbst auf dem Bildschirm erscheint und nur die Meldung des COPY-Befehls

(xxx Datei(en) kopiert)

auf dem Drucker. Denn die Ausgaben des COPY-Befehls (und nur die sind umzulenken) bestehen aus den jeweiligen Ausschriften – nicht aus dem Ziel des Kopiervorganges, der eine interne Aktion des Befehls ist. So würde

copy datei > prn

auch nur die Fehlerausschrift „Datei kann nicht auf sich selbst kopiert werden" auf den Drucker bringen.
Probieren Sie am besten noch ein wenig.

Filter

Über die Umleitung von Ein- und Ausgabe hinaus kann der Datenstrom auch durch ein anderes Programm hindurchgeleitet werden. Dazu werden die Ausgaben des einen Programms einfach als Eingaben des anderen betrachtet. Diese Programme heißen Filter (englisch Pipes) und werden mit dem Ziel benutzt, die Ausgabe in irgendeiner Weise zu verändern, eben zu filtern. Zum Lieferumfang von DOS gehören drei Filter: SORT, FIND und MORE. Darüber hinaus kann man sich aber auch selbst Filterprogramme schreiben. Am einfachsten zu gebrauchen ist MORE. Es gibt stets nur 23 Zeilen auf den Bildschirm aus und wartet dann auf einen Tastendruck. Zum Anzeigen eines langen Directories verwendet man

dir | more

Wie zu sehen, dient zur Filterung eines Datenstromes der senkrechte Strich (|). Wenn Sie damit Probleme auf der Tastatur haben, so erinnern Sie sich bitte der Nutzung des ASCII-Codes. Mit dem Wissen, daß der Code für den Strich 124 (dezimal) ist, können Sie ihn erzeugen, indem die ALT-Taste gedrückt, rechts 124 getippt und die ALT-Taste wieder losgelassen wird. Mit dem Filterzeichen werden die Ausgaben des DIR-Befehls als Eingaben von MORE weitergeleitet. Das MORE-Filter erlaubt so eine seitenweise Ausgabe. Noch weniger als für den DIR-Befehl (wo man ja den /p-Schalter hat) ist MORE für TYPE:

type datei | more

Das SORT-Filter sortiert den Datenstrom. Zur alphabetisch sortierten Ausgabe des Verzeichnisses schreiben Sie

dir | sort

Es geht auch absteigend mit

dir | sort /r

Es muß nicht unbedingt nach dem Anfangsbuchstaben sortiert werden; die Anweisung

dir | sort /+9

sortiert die Dateien nach dem Typ, der gerade in der 9. Spalte einer jeden Zeile steht. Beachten Sie bitte, daß damit nur die Anzeige auf dem Bildschirm geändert wird, nicht etwa die Reihenfolge der Einträge in das Inhaltsverzeichnis.
Durch das SORT-Filter können auch Dateien geschickt werden. Mit

sort < datei

erfolgt eine sortierte Ausgabe auf den Bildschirm, und mit

sort < altdatei > neudatei

wird das Ergebnis wiederum in eine Datei geschrieben. Probieren Sie es mal mit einer beliebigen Textdatei.
Das Sortieren der oben angelegten Datei INHALT mit den Verzeichnissen der Laufwerke A:, B: und C: würde diese Verzeichnisse übrigens bunt durcheinandermischen. Zum Schluß überlegen Sie bitte einmal, warum bei den letzten Befehlen das Filter am Anfang der Kommandozeile stand und nicht einfach **dir > sort** geschrieben werden kann. Sie können es gefahrlos ausprobieren: Es wird eine Datei mit Namen SORT angelegt, und das war wohl nicht Sinn der Sache.
Noch vielfältiger kann das FIND-Filter eingesetzt werden, das ungefähr das macht, was man von einem Filter erwartet: Es filtert bestimmte Teile heraus. Gearbeitet wird zeilenweise, das heißt, es wird die gesamte Zeile ausgegeben, die die Suchbedingung erfüllt. So werden mit

dir | find "BAT"

nur die Dateien angezeigt, die die Zeichenkette "BAT" enthalten. Nicht eben genial, das hätte man auch mit Jokern erreicht. Aber können Sie auch alle Dateien außer den BAT-Dateien anzeigen?
Kein Problem:

dir | find /v "BAT"

Achten Sie bitte darauf, daß an dieser Stelle Großbuchstaben für die Zeichenkette zu verwenden sind, da die Eintragungen im Directory sämtlich in Großbuchstaben umgewandelt werden. Ähnlich müssen Sie verfahren, wenn Sie nur die Unterverzeichnisse anzeigen wollen:

dir | find "<"

Oder das Anzeigen aller Dateien, die im September erstellt wurden:

dir | find ".09."

Wichtig ist nur, eine signifikante Zeichenkette zu finden, die die entsprechenden Zeilen von allen anderen unterscheidet.
Statt als Filter kann man FIND auch als eigenständiges Programm verwenden, etwa

um die Zeilen dieses Kurses zu zählen, in denen das Wort „DOS" vorkommt:

find /c "DOS" dos4.doc

Die Antwort ist

—————— dos4.doc: 14

Bis jetzt kam also in 14 Zeilen das Wort „DOS" vor, was nicht unbedingt bedeutet, daß es nur 14mal im Text Verwendung fand, da es ja auch mehrmals auf einer Zeile stehen kann und dann nur einmal gezählt wird. Es wird – wie schon erwähnt – zeilenweise gearbeitet.

```
FILTER

Seitenweise Ausgabe
MORE

Sortieren
SORT [/R] [/+n]
/R     umgekehrte Sortierreihenfolge
/+n    sortieren nach Spalte n

Suchen nach Zeichenketten
FIND [/C] [/N] [/V] "string"
/C      Zählen der Zeilen, die die
        Zeichenkette enthalten
/N      Anzeige mit Zeilennummer
/V      Suchen aller Zeilen, die die
        Zeichenkette nicht enthalten
```

Bild 2

Durch Leerzeichen getrennt können auch mehrere Dateien aufgeführt werden, die dann nacheinander durchmustert werden; der Einsatz von Jokern für Dateiengruppen ist aber nicht erlaubt.

Niemand hindert Sie, ihre Daten durch beliebig viele Filter zu schicken, beispielsweise um alle Dateien, die nicht die Erweiterung DOC haben, nach Namen sortiert seitenweise auf den Bildschirm auszugeben:

dir | find /v "DOC" | sort | more

Sie werden bemerken, daß die Ausführung etwas länger dauert, denn DOS baut intern Hilfsdateien auf, in denen gesucht und sortiert wird. Diese temporären Dateien werden automatisch gelöscht, können aber unter Umständen bei einem Absturz oder einem Reset während der Abarbeitung auf der Festplatte zurückbleiben. Der Name dieser Dateien mutet seltsam an (z. B. 0e14081a), weil er bei neueren Betriebssystemversionen aus der Zeit gebildet wird, um im Netzbetrieb nicht zu kollidieren.

Denken Sie nur nicht, Filter seien eine komplizierte Sache. Jeder kann sich mit einfachen Mitteln selbst welche schreiben. Diese Programme müssen den Datenstrom von der Standardeingabe einlesen, ihre Aufgabe ausführen und wieder zur Standardausgabe schreiben. Sehen sie, wie kurz ein Pascal-Programm ist, das – als Filter verwendet – Texte mit Zeilennummern versieht:

```
program number;
var zeile : string;
    i     : integer;
begin
  i:=1;
  while not eof do begin
  readln(zeile);
  writeln(i:3,'      ',zeile);
  i:=i+1;
  end;
end.
```

Durch dieses Filter können jetzt beliebige Texte geschickt werden, z. B. das Inhaltsverzeichnis mit

dir | number

oder auch dieser Kurs:

type dos4.doc | number

Es ist auch kein Problem den Text mit Zeilennummern wieder abzuspeichern:

number < dos4.doc > dos4num.doc

Mit Hilfe des Kleiner- und Größerzeichens wird der Ein- und Ausgabestrom durch das Filter geleitet. Unter dem Namen „dos4num.doc" steht der numerierte Text wieder auf der Platte. Auf diese Art und Weise können Sie sich ganz leicht beliebige Filter selbst schreiben. Weitergehende Informationen zu Filtern finden sich in MP 9/89, S. 263.

Filter können weiterhin dazu verwendet werden, Rückfragen von Befehlen zu begegnen. Wenn am Ende einer Sitzung nach dem Sichern stets alle Dateien gelöscht werden sollen, so erzeugt der Befehl **"del *.*"** stets die Ausschrift **"Wirklich löschen? j/n"**, was in einer BAT-Datei lästig sein kann. Mit

echo j | del *.*

wird nicht angehalten. Etwas schwerer ist es schon, Befehle zu bedienen, die nur ein <ENTER> erwarten, zum Beispiel FORMAT. Der Befehl

echo.| format a:

beginnt mit dem Formatieren, ohne auf Tastendruck zu warten. Zwischen Punkt und senkrechtem Strich darf kein Leerzeichen stehen. Auch DATE und TIME warten stets auf das Drücken der ENTER-Taste, wenn Datum oder Zeit nur angezeigt werden sollen. Kein Problem mit

echo.| time

Und wenn nur die Zeile mit der Zeit gefragt ist, so filtert man den Rest einfach weg:

echo.| time | find "."

Natürlich gibt das nur in Batchdateien richtig Sinn. Ein <ENTER> kann übrigens auch mit dem MORE-Filter erzeugt werden. Damit wird ausgenutzt, daß es stets eine Leerzeile einfügt. Statt des oben beschriebenen Befehls kann so auch

echo | more | time

geschrieben werden.

Wie Sie sehen, muß man manchmal schon ganz schöne Klimmzüge veranstalten, um elegante Lösungen zu erreichen, aber die DOS-Kommandosprache ist halt recht klein – dafür aber leicht zu erlernen.

Ihnen wird aufgefallen sein, daß immer häufiger von Dienstprogrammen die Rede ist. Diese haben heute die meisten DOS-Befehle verdrängt, die hier – um ein solides Grundwissen zu vermitteln – ausführlich beschrieben wurden. In der nächsten Folge soll deshalb ein Programm im Mittelpunkt stehen, das sich – ob als Dienstprogramm oder als Nutzeroberfläche – beachtlicher Beliebtheit erfreut: der Norton-Commander.

Neuerungen von MS-DOS 4.x

Obwohl die neueste MS-DOS-Version die Nummer 4.x trägt, wird in der Praxis überwiegend noch mit Version 3.3 gearbeitet. Was bietet ein Umstieg für Vorteile? Auffallendste äußere Neuerung ist die zum Betriebssystem gehörige grafische Benutzeroberfläche (DOS-Shell genannt), die aber – verglichen mit Windows der GEM – recht primitiv wirkt und sich auch mit beliebten Shells auf alphanumerischer Basis (Norton Commander, PC Tools) nicht messen kann.

Die wirklich gravierende Neuerung besteht in der Nutzung von Festplatten mit mehr als 32 MByte als ein logisches Laufwerk. Unter DOS 3.x ist eine größere Festplatte in mehrere Laufwerke aufzuteilen (zu partitionieren). Angesichts ständig billiger werdender großer Festplatten mag dieser Umstand ein echter Beweggrund für einen Umstieg auf eine neue Version von MS-DOS sein – allerdings um den Preis, daß einige hardwarenahe Programme nicht mehr korrekt arbeiten. Probleme wurden mit den Norton Utilities und leider auch mit dem neuen Wordperfekt 5.1 festgestellt. Ursache dafür ist die Erweiterung einer DOS-internen Tabelle, des DOS-Parameter-Blocks (DPB), der die nähere Beschreibung der Datenträger enthält. Die alte Beschränkung auf 32 MByte ergibt sich übrigens aus der Sektoradressierung mit einer 16-Bit-Zahl. 65 536 Sektoren zu je 512 Byte sind so erreichbar, was einer Kapazität von 32 MByte entspricht. Von untergeordneter Bedeutung dürfte die Unterstützung des Expanded Memory durch DOS 4.x sein, da die für den 8086 eingeführte Hardware auf ATs kaum genutzt wird. Mit dem Treiber XMA2EMS.SYS läßt sich EMS als Datenbereich oder Cachepuffer einrichten (beispielsweise läßt sich der mit *Buffers* reservierte Speicherbereich auslagern), VDISK.SYS unterstützt eine RAM-Disk im EMS. Wichtiger noch mag der Treiber XMAEM.SYS sein, der EMS im Extended Memory (wie ihn viele ATs besitzen) emuliert.

Vier Befehle sind bei DOS 4.x neu hinzugekommen:

DOSSHELL	– Start der grafischen Nutzeroberfläche
INSTALL	– Installation von DOS-Programmen bereits in Config.SYS
MEM	– Zeigt die Hauptspeicherbelegung detailliert an, wozu bisher Hilfsprogramme (Mapmem) notwendig waren.
SWITCHES	– Schaltet die erweiterten Tastaturfunktionen ab (einige Programme erfordern das).

Zusätzlich wurde der Funktionsumfang einer Reihe vorhandener Befehle erweitert. Beispielsweise zeigt *Tree* das Verzeichnissystem jetzt wirklich als Baum.

Übrigens: Für Ende dieses Jahres hat Microsoft – entgegen ursprünglichen Absichten – eine grundlegend neue Version von MS-DOS angekündigt, die voraussichtlich die Nummer 5.0 tragen wird.

Einführung in MS-DOS

Teil 5

Wolfram Schulze, Uwe Schulze, Berlin

Der Norton Commander

Nicht alles ist so schwer, wie in den bisherigen vier Folgen beschrieben – es gibt ja Dienstprogramme. Heute vorgestellt: Der Norton Commander.

Dienstprogramme und Nutzeroberflächen

In den erster vier Folgen unseres DOS-Kurses wurden die Kommandos in sehr konservativer Weise behandelt. Nun hebt sich aber gerade DOS von anderen Betriebssystemen dadurch ab, daß eine Reihe komfortabler Bedienoberflächen, Dienstprogramme, Fullscreen-Editoren und Tools aller Couleur verfügbar sind. Zumindest drei Argumente sprechen aber für eine ausführliche Behandlung der DOS-Kommandooberfläche: Erstens läßt sich nur so ein echtes Verständnis für die innere Struktur von DOS vermitteln. Zweitens kann man die Befehle in Batch-Dateien zusammenfassen oder auch in Programmen verwenden (indem der Kommandointerpreter ausgeführt wird). Und drittens: Was tun Sie, wenn einmal ein Dienstprogramm zur Hand ist?

Nun war von einfacher Bedienung (die die weite Verbreitung von MS-DOS gefördert haben soll) an mancher Stelle der vorangegangenen Kurs-Folgen nichts zu merken – denken Sie nur an die Batch-Dateien. Dieser Mangel an Bedienkomfort war wohl auch den Betriebssystem-Herstellern bekannt. Sie lieferten nachträglich grafische Nutzeroberflächen (Windows, GEM), die insbesondere dem Einsteiger den Kommandodschungel ersparen sollen. Bei den neuen Betriebssystemen wird der Befehlsinterpreter von einer menügeführten Nutzeroberfläche (Shell) abgelöst (DOS-Shell bei MS-DOS 4.0, Presentation Manager bei OS/2). Ihr Vorteil liegt in einem einheitlichen äußeren Erscheinungsbild, an das sich auch immer mehr andere Software anlehnt und das IBMs SAA (Systems Application Architecture) entspricht. In jedem Falle werden aber ein grafikfähiges Gerät und eine Maus benötigt.

Neben den erwähnten Shells existieren noch eine Reihe von Dienstprogrammen (Pctools, Qdos, Xtree), die anstelle der internen und externen Systembefehle eingesetzt werden und die statt langer Kommandofolgen nur weniger Handgriffe bedürfen. Auf eine Gegenüberstellung dieser Programme wurde hier verzichtet – jeder mag verwenden, was er am besten beherrscht und was den Anforderungen entgegenkommt. Statt dessen wird der Norton Commander – als ein typischer Vertreter – genauer vorgestellt, da er weite Verbreitung erlangt hat und viele nützliche Eigenschaften der oben genannten Bedienoberflächen und Dienstprogramme in sich vereint.

Die Bedienung des Norton Commanders

Der Start des Norton Commanders kann auf zwei Wegen erfolgen. Ruft man Nc.exe auf, so wird das gesamte Programm resident in den Speicher gelegt, und das kostet immerhin 140 KByte Hauptspeicher. Da sich viele Anwendungen darauf gar nicht erst einlassen (in Turbo-Pascal kann man nur noch wenige Zeilen große Programme übersetzen), wird man sich zumeist für die zweite Variante entscheiden. Dazu installiert man mit Ncsmall.-exe ein kleines residentes Programm, das nur etwa 12 KByte Speicher permanent belegt und den Rest des Norton Commanders nachlädt – diesen Speicherplatz aber freigibt, wenn ein anderes Programm gestartet wird. Eine solche Verfahrensweise ähnelt der bei Command.com, wo auch ein residenter und ein transienter (nachzuladener) Teil unterschieden werden. Der Nachteil liegt auf der Hand: Das Nachladen kostet Zeit. Hier kann man sich wiederum helfen, indem man Nc.exe auf eine RAM-Disk kopiert.

Ein Blick auf die Festplatte (oder Diskette) zeigt Ihnen, daß noch weitere Dateien zum Norton Commander gehören. Sie dienen insbesondere der Anpassung an die individuellen Wünsche des Nutzers. Im einzelnen können folgende Files vorhanden sein:

Nc.ini
Nc.ext (Textdatei)
Nc.mnu (Textdatei)
Treeinfo.ncd

Später werden wir sehen, wie diese Dateien modifiziert (oder, falls noch nicht vorhanden, angelegt) werden. In die Textdateien kann aber schon jetzt mit einem Editor oder mit Type ein Blick geworfen werden. Die ini-Datei (ini für Initialisation) enthält Angaben zur Konfiguration, betrifft also das Outfit, und das kann recht unterschiedlich sein. In Nc.ext (ext für Extension) sind Kommandos eingetragen, die in Abhängigkeit von der Dateierweiterung ausgeführt werden, Nc.mnu (mnu für Menü) beinhaltet nutzerdefinierte Menüs und in Treeinfo.ncd merkt sich Norton den Verzeichnisbaum, was insbesondere bei großen Festplatten zu erheblicher Zeitersparnis führt. Im folgenden wird auf die weit verbreitete Version 2.0 Bezug genommen, inzwischen wird bereits die Folgeversion 3.0 geliefert. Dort trägt das Hauptprogramm den Namen NCMAIN.EXE, und der residente Teil heißt NC.EXE, so daß ein NCSMALL entfällt.

Beim Aufruf erscheint das Programm wie in Bild 1 – in Abhängigkeit von den Eintragungen in Nc.ini, aber auch ein wenig anders. Zwei Verzeichnisfenster bedecken fast den gesamten Bildschirm. Sie werden Panels (engl., = Tafel, rechteckiges Format) genannt und enthalten die Datei- und Verzeichniseinträge. In einem Fenster befindet sich ein Balkencursor, der mit den Pfeiltasten auf und ab bewegt werden kann. Stellt man ihn auf den obersten oder untersten Eintrag, so wird automatisch gescrollt (Kunstwort aus screen und rolling), also die Einträge werden um eine Position nach unten oder oben verschoben. So sind immer alle Dateieinträge zu erreichen. Weiterhin ist das Blättern mit <PgUp> bzw. <Bild ↑> und <PgDn> bzw. <Bild ↓> sowie das direkte Anwählen des ersten Eintrages mit <Home> bzw. <Pos 1> und des letzten mit <End> bzw. <Ende> möglich.

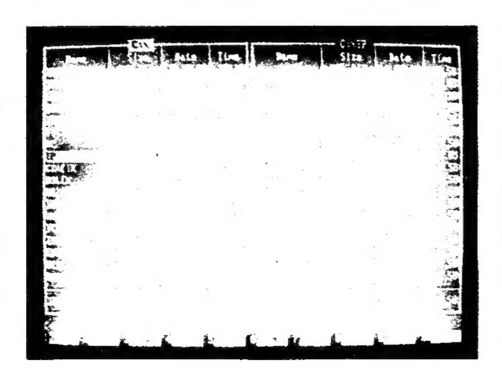


Bild 1 In dieser Einstellung belegt der Norton Commander den gesamten Bildschirm

Ein großer Vorteil des Norton Commanders besteht darin, daß auch die traditionelle Kommandozeile jederzeit zur Verfügung steht. In Bild 1 sehen Sie auf der zweiten Zeile von unten den DOS-Prompt, und dahinter steht der blinkende Cursor. Es ist also kein Problem, einmal ein **dir** einzugeben. Dann passiert genau das, was man erwartet – für die Zeit der Ausführung verschwinden aber die Verzeichnisfenster des Commanders. Und so hält er es immer, wenn andere Programme gestartet werden. Den DIR-Befehl wird man wahrscheinlich nicht mehr verwenden, aber probieren Sie doch einmal einen Laufwerkwechsel. Dazu ist einfach **a:** einzugeben (sofern auch eine Diskette eingelegt ist). Sie werden feststellen, daß anschließend auch die Datei von Laufwerk A: im aktuellen Panel angezeigt werden. DOS und der Norton Commander arbeiten also Hand in Hand. Zum Wechseln des Laufwerkes kann man aber auch eine Tastenkombination verwenden, nämlich <Alt>-<F1> für das linke und <Alt>-<F2> für das rechte Fenster. Es erscheint dann ein kleines Menü, aus dem das neue Laufwerk ausgewählt werden kann. Das geht auch nicht schneller als in der Kommandozeile, hat aber den Vorteil, daß alle vorhandenen logischen Laufwerke (also auch mit Supercopy eingerichtete oder im Netzwerk angemeldete) aufgelistet werden.

In den beiden Panels sind nicht nur unterschiedliche Laufwerke einstellbar, sondern auch unterschiedliche Verzeichnisse eines Laufwerks. Traditionell kann dazu **cd** für Change Directory verwendet werden, einfacher ist es aber, den Balkencursor auf den entsprechenden Directory-Eintrag zu positionieren und <Enter> zu drücken. Verzeichnisse sind im Gegensatz zu Dateien an den Großbuchstaben zu erkennen; außerdem sind sie mit **SUB-DIR** gekennzeichnet. Wie gehabt stehen für das übergeordnete Verzeichnis zwei Punkte. Im linken Panel von Bild 1 gibt es kein übergeordnetes Verzeichnis; wir befinden uns im Hauptverzeichnis, was auch an der Pfadangabe in der obersten Zeile zu sehen ist. Der Prompt der Kommandozeile zeigt das Verzeichnis des aktuellen Fensters (zu erkennen am Balkencursor) an. Wechseln Sie doch einmal mit <Ctrl>-<I>

oder der Tabulatortaste in das andere Verzeichnisfenster und achten Sie auf den Prompt. Anschließend sollten Sie versuchen, ein anderes Directory einzustellen (cd bleibt Ihnen dafür immer), indem Sie den Balkencursor auf einen Directory-Eintrag setzen und mit <Enter> abschließen.
Die Directories stehen stets vor den Dateien im Fenster, so daß man mit <Home> bzw. <Pos1> schnell nach oben blättern kann. Da Verzeichnisoperationen sehr häufig benutzt werden, gibt es noch eine Reihe von Tastenkombinationen (Shortcuts) zum schnellen Wechsel. So gelangt man mit <Ctrl>-<PgUp> bzw. <Strg> <Bild↑> von jeder Stelle im Panel in das übergeordnete Verzeichnis, und mit <Ctrl>-<\> bzw. <Strg>-<\> wechselt man in das Hauptverzeichnis (Root) (\ bezeichnet dabei die Taste, nicht das Zeichen).
Es gibt noch eine weitere Möglichkeit, schnell zu einer bestimmten Datei im aktuellen Verzeichnis zu gelangen. Dazu tippen Sie <Alt> und den Anfangsbuchstaben ein. Der Balkenkursor springt auf den ersten Eintrag dieses Buchstabens. Die Angabe der folgenden Buchstaben (sofern Sie das wollen) kann auch ohne <Alt> erfolgen. Ein kleines Suchfenster am unteren Panelrand hält Sie darüber auf dem laufenden, ob überhaupt Dateinamen mit dieser Buchstabenfolge vorhanden sind.
Nun soll die Bedienung einer Nutzeroberfläche nicht nur schnell von der Hand gehen, sondern auch komfortabel sein. Und wer die Auswahl aus einem grafisch dargestellten Dateibaum (z. B. in Pctools) kennt, braucht auch hier darauf nicht zu verzichten. Die Tastenkombination <Alt>-<F10> liefert einen solchen – der Nutzer kann mit den Cursortasten auswählen. Es sollte also ab jetzt kein Problem mehr sein, in beiden Panels jeweils das Laufwerk und den Pfad Ihrer Wahl einzustellen.
Zum Starten von Programmen brauchen Sie nur – sofern Sie nicht den Aufruf in der Kommandozeile bevorzugen – den Balkencursor auf dem Dateinamen zu positionieren und <Enter> zu drücken. Für die Zeit der Ausführung ist der Norton Commander nicht verfügbar, anschließend meldet er sich – wie oben beschrieben – wieder. Sollen in der Kommandozeile noch weitere Angaben gemacht werden (z. B. Parameter übergen), so kopiert man den Dateinamen einfach mit <Ctrl>-<Enter> in die Kommandozeile und ergänzt dann.

Funktionen

Der Norton Commander verfügt über einen integrierten Editor. Man braucht nur auf einer Textdatei zu positionieren und ist mit der Funktionstaste <F4> im Editiermodus. Die einfachen Funktionen sind Wordstar-kompatibel, leider existieren die Blockoperationen aber nicht. Einzig eine Suchoperation ist vorhanden (Funktionstaste <F7>). Die nicht in der Hilfszeile aufgeführte Suchwiederholung mit dem gleichem Begriff erreichen Sie mit <Shift>-<F7>. In der Kopfzeile werden die Zeilen, Spalten und freier Speicherplatz angezeigt. Dateien bis zu einer Größe von 30512 Byte können unter MS-DOS 3.30 und einem Hauptspeicher von 640 KByte bearbeitet werden. Ein kleines Extra finden Sie an der äußersten rechten Position, wenn diese nicht durch ein residentes Clock-Programm überdeckt ist: den ASCII-Wert des Zeichens, auf dem der Kursor gerade steht bzw. die Abkürzungen für End of Line und End of File. Das Sternchen zeigt an, ob am Text Änderungen vorgenommen wurden und ob deshalb beim Verlassen ein eventuelles Sichern hinterfragt werden muß. Eine Unterteilung in Seiten erfolgt nicht – Ziel ist die Programmerfassung und nicht die Textverarbeitung.
Die meisten Funktionen erschließen sich dem Nutzer leicht oder sind eh bekannt. Deshalb sei es erlaubt, auch auf ein paar Optionen einzugehen, die nicht so geläufig sind. Zum Beispiel darauf, daß es möglich ist, im integrierten Editor auch nicht druckbare Zeichen zu verwenden. Dazu ist <Ctrl>-<Q> einzugeben, und anschließend kann wie auf Kommandoebene mit <Alt> und dem Kode auf dem Ziffernblock das entsprechende Steuerzeichen erzeugt werden. In diesem Zusammenhang sei an die Schwierigkeiten bei der Verwendung von ANSI-Sequenzen in den letzten Kurs-Folgen erinnert. Und noch etwas: Viele Nutzer greifen immer noch auf **copy con datei** zurück, wenn sie eine neue Datei anlegen wollen. Versuchen sie es statt dessen einmal mit <Shift> <F4>. Norton fragt dann nach einem Dateinamen und erzeugt – falls nicht vorhanden – eine neue Datei. Ist Ihnen der Build-in-Editor zu spartanisch ausgestattet, so kann statt dessen ein beliebiger Ersatz (z. B. der Norton Editor oder Wordstar) – wie später bei **Setup** gezeigt – integriert werden.
Neben der Edit-Funktion (siehe auch Hilfe auf der untersten Bildschirmzeile) gibt es noch **View** zum Ansehen von Dateien (Funktionstaste <F3>). Es erlaubt die Anzeige weit größerer Dateien, als der Editor aufnehmen könnte und bewahrt Sie vor versehentlichen Änderungen. Verlassen wird dieser Modus – wie auch der Editor – mit <ESC>.
Neben der obligaten Hilfe auf <F1> zeigt die KeyBar genannte untere Bildschirmzeile noch Funktionen zum Kopieren, Umbenennen und Löschen. Sie beziehen sich im einfachsten Fall auf die Datei, auf der der Balkencursor gerade steht. Drücken Sie also <F5>, so wird nach dem Ziel des Kopiervorganges gefragt. Vorgabe ist stets das im anderen Fenster eingestellte Verzeichnis, sie kann aber beliebig abgeändert werden. Beachten Sie dabei bitte folgendes: Wird als erstes eine Bewegung mit den Kursortasten ausgeführt, so bleibt die Vorgabe stehen und kann verändert werden; schreiben Sie zuerst ein Zeichen, so verschwindet die Vorgabe ganz, und es wird eine komplette neue Eingabe erwartet. Die Angabe des Kopierzieles unterliegt den gleichen Regeln wie beim COPY-Befehl.
Mit <F6> erhalten Dateien einen neuen Namen. Weiterhin ist ein Verschieben möglich, sofern ein anderes Zielverzeichnis festgelegt wird. Verschieben bedeutet in diesem Falle Kopieren und anschließendes Löschen der Quelldatei, auch beim Verschieben von Diskette auf Festplatte und umgekehrt, was mitunter sehr hilfreich aber auch verhängnisvoll sein kann. Bei Löschen mit <F8> ist zu beachten, daß auch schreibgeschützte Dateien nach Rückfrage und Bestätigung mit der Enter-Taste gelöscht werden. Ein wenig Aufmerksamkeit ist also vonnöten.
Wie der Hilfszeile zu entnehmen ist, werden mit <F7> neue Verzeichnisse angelegt, aber das ist mit **md** wohl genauso schnell zu bewerkstelligen.
Der Norton Commander bietet auch die Möglichkeit, Dateien in Verzeichnissen zu behandeln, in denen man sich gerade nicht befindet. Dazu sind die eben beschriebenen Funktionen zusammen mit <Shift> zu verwenden.Wenn Sie <Shift>-<F5> drücken, so werden Sie nach Quell- *und* Zieldatei gefragt. Desgleichen auch für das Umbenennen und Löschen und – wie oben bereits erwähnt – für das Editieren von Dateien.
Noch gar nicht erklärt wurde, wie man mehrere Dateien zugleich behandelt. Dazu müssen diese markiert werden. Das geschieht mit <Ins> bzw. <Einfg> für jeweils die Datei, auf der der Balkencursor steht. Auf einem Farbdisplay ist die Markierung gut zu erkennen, im Schwarz/Weiss-Modus wird sie durch erhöhte Helligkeit oder durch einen Balken neben dem Dateinamen angezeigt. Mit erneutem Drücken von <Ins> kann – wenn der Balken auf der markierten Datei steht – die Markierung wieder aufgehoben werden. Eine weitere Erleichterung stellt die Markierung mit einer Dateimaske und die Verwendung der Joker dar. Dazu drücken Sie die Plus-Taste neben dem rechten Ziffernblock (zum Aufheben der Markierung die darüber liegende Minus-Taste) und es erscheint eine Maskenvorgabe, in diesem Fall *.* für alle Dateien. Es steht Ihnen frei, statt dessen eine beliebige Dateigruppe einzutragen. Anschließend kann für alle diese Datei eine Aktion ausgeführt werden.
Besonders hilfreich ist die Arbeit mit markierten Dateien, wenn auf mehrere Disketten kopiert werden soll. Ist eine Diskette voll, so ist ein Wechseln möglich, denn nur die verbleibenden Dateien sind noch markiert. Wollen Sie dagegen einmal ein ganzes Verzeichnis löschen, das noch dazu sehr viele Dateien enthält, so sind Sie gut beraten, dies mit del *.* zu tun, da eine solche Aktion im Norton Commander – bedingt durch das einzelne Anzeigen jeder Aktion – recht viel Zeit in Anspruch nimmt.
Immer wieder sucht man auf der Festplatte nach Dateien, von denen man nicht genau weiß, in welchem Verzeichnis sie stehen. Mit <Alt>-<F7> kann veranlaßt werden, nach einer Datei oder einer Gruppe von Dateien (bei Angabe einer Maske) auf dem aktuellen Laufwerk zu suchen. Alle gefundenen Dateien werden in einem Fenster angezeigt und können mit den Cursortasten ausgewählt werden.
Wer oft auf der Kommandoebene arbeitet, wird es zu schätzen wissen, in der Kommandozeile zurückblättern zu können – eine Option, die Ihnen vielleicht aus dBase oder vom Public-Domain-Programm Dosedit geläufig ist. Leider sind die Pfeiltasten bereits zur Bewegung des Balkenkursors im Verzeichnisfenster vergeben. Sie können aber mit der Ctrl-bzw. Strg-Taste die Tastenkodes erzeugen, die denen der Pfeiltasten entsprechen, nämlich <Ctrl>-<E> für aufwärts und <Ctrl>-<X> für abwärts. Mit diesen beiden Tastenkombinationen ist es möglich, im

Kommandostapel zu blättern, das heißt die zuletzt eingegebenen Befehle wieder in die Kommandozeile zurückzuholen und gegebenenfalls zu korrigieren. Zusätzlich bewahrt der Norton Commander die letzten Kommandoeingaben in einem separaten Fenster (History genannt) auf. Sie können es mit <Alt> <F8> aktivieren und anschließend menügeführt auswählen. Leider sind keine Veränderungen am Befehl mehr möglich.
Häufig verdeckt der Norton Commander wichtige Teile des Bildschirms. Abhilfe schafft die Möglichkeit, den gesamten Commander oder einzelne Teile auszublenden. MIt <Ctrl>-<F1> läßt sich das linke mit <Ctrl>-<F2> das rechte Panel aus- und wieder einblenden. Gleiches gilt für die Hilfszeile am unteren Bildschirmrand mit <Ctrl>-<B>. Für die meisten Aktionen ist ein Verzeichnisfenster ausreichend, das man zweckmäßigerweise auf der rechten Seite stehen läßt und so freien Blick auf die Ausgaben der gestarteten Programme hat.
Die Tastenkombination <Ctrl>-<O> blendet beide Panels aus. Beachten Sie aber bitte, daß der Norton Commander nicht nur im Speicher, sondern auch weiterhin aktiv bleibt, wie Sie am Betätigen der Funktionstasten leicht erkennen können. Zusätzlich kann jetzt auch mit den Pfeiltasten in der Kommandozeile geblättert werden.
Zu beachten ist, daß der Norton Commander beim Laden nicht prüft, ob er sich bereits im Speicher befindet. Er läßt sich also leicht ein zweites Mal laden, wenn Sie ihn nach <Ctrl> <O> erst einmal aus den Augen verloren haben. Spätestens beim vierten Mal ist dann aber der 640-KByte-Speicher voll.
Nicht vergessen wollen wir zu guter Letzt, wie man den Commander wieder aus dem Speicher entfernt. Wie anfangs erwähnt, wird es in erster Linie Speicherplatzmangel sein, der Sie dazu bewegt. Beim Verlassen mit <F10> wird noch einmal rückgefragt, um ein versehentliches Betätigen der Taste korrigieren zu können. Beachten Sie, daß alles, was Sie eingestellt haben, damit verlorengeht. Es sei denn, die Konfiguration wird mit <Shift>-<F9> in die Datei Nc.ini gespeichert. In diesem Falle findet der Nutzer beim erneuten Start alles so vor, wie es beim letzten Mal verlassen wurde. Wurde ein Fenster oder die Hilfszeile ausgeblendet, so wird auch das berücksichtigt.

Konfigurieren (Setup)

Der Norton Commander bietet weiten Freiraum zur Anpassung an die eigenen Wünsche und Erfordernisse. Das betrifft in erster Linie das Outfit, also die äußere Erscheinung – in begrenztem Umfange aber auch die Funktionalität. Sämtliche Einstellungen werden über das Setup-Menü vorgenommen, das Sie mit <F9> erreichen. Die am oberen Bildschirmrand erscheinende Menüzeile enthält 5 Einträge, die für jeweils ein Pull-down-Fenster stehen. Die beiden äußeren Menüs sind für die Einstellung der Panele zuständig (siehe Bild 2). Mit den oberen fünf Einträgen wird festgelegt, was im Verzeichnisfenster angezeigt werden soll. **Brief** stellt die Anzeige der Dateien in Kurzform (ohne Größe und Zeit) ein, wie in Bild 3 rechts zu sehen. Hinter **Full** verbirgt sich die vollständige Anzeige, die wir aus Bild 1 kennen. Statt der Verzeichniseinträge kann auch der Verzeichnisbaum grafisch dargestellt werden, wie in Bild 3 links zu sehen. Dazu ist **Tree** zu wählen. Weiterhin können mit **Info** Informationen zum Hauptspeicher (gesamt, belegt) und zum aktuellen Laufwerk (Kapazität, frei) zur Anzeige gebracht werden. Da man diese nicht ständig benötigt, gibt es einen Shortcut zum zeitweiligen Umschalten, nämlich <Ctrl>-<L>. Warum ist diese Tastenkombination im Setup-Menü nicht aufgeführt?
Die mittleren fünf Punkte in Bild 2 betreffen die Reihenfolge, in der die Dateien angezeigt werden sollen: Sortiert nach Namen, Typ, Zeit und Größe. Beachten Sie aber, daß diese Sortierung nur die Anzeige betrifft und nicht etwa die Änderung der Einträge im Verzeichnis der Diskette, wie das beispielsweise Pctools tut. Die Tastenkombinationen <Ctrl>- <F1> zum Ausschalten des Panels sowie <Alt>-<F1> zum Wechseln des Laufwerks im linken Panel haben wir bereits kennengelernt, für das rechte Panel gilt entsprechend <F2>, was Sie auch im rechten Pull-down-Menü einstellen können.
Die Option **Re-read** veranlaßt, daß nach einem Diskettenwechsel die Einträge im Verzeichnisfenster aktualisiert werden. Das gleiche erreicht man aber schneller durch nochmaliges Anwählen des Laufwerks mit **a:** <enter> oder mit <Ctrl>-<R>.
Die Version 3.0 kennt darüber hinaus eine Filter-Funktion, die nur die durch eine Maske vorgegebenen Dateien anzeigt. Der Unterpunkt **Files** im Setup-Menü erlaubt noch einmal das interaktive Auswählen aller Funktionen, die auch in der Hilfszeile stehen und die wir bereits besprochen haben. Ein paar zusätzliche Einstellungen finden wir aber im **Option**-Menü (siehe Bild 4). In diesem Menü werden Schalter gestellt, die jeweils zwei Zustände annehmen können, was an einem kleinen Häkchen zu erkennen ist. Mit Color kann zwischen Farb- und Monochromdisplay unterschieden werden. Die Erfahrung zeigt, daß für eine Reihe von Schwarzweiß-Bildschirmen die Festlegung als Laptop die besten Kontraste bringt. Das betrifft besonders die Darstellung der markierten Dateien.
Der Punkt **Auto menus** veranlaßt, daß beim Start des Norton Commanders stets das nutzereigene Menü automatisch aktiviert wird, welches sonst unter <F2> erreichbar ist. Eine Maßnahme besonders für unbedarfte Nutzer.
Mit **Path prompt** kann die Anzeige des aktuellen Pfades im Prompt ein- und ausgeschaltet werden. Man kann auf die Pfadanzeige durchaus verzichten, da an der Kopfzeile jedes Panels das aktuelle Verzeichnis ablesbar ist. Das Ausblenden der Hilfszeile (KeyBar) durch <Ctrl>-<B> ist Ihnen bereits vertraut. Die Verzeichnisfenster können auf den ganzen Bildschirm ausgedehnt (wie in Bild 1) oder nur auf halbe Höhe gezogen werden (wie in Bild 3). Dafür ist der Punkt **Full Screen** zuständig. **Mini status** bezieht sich auf die unterste Zeile der Panels in den Bildern 1 und 3. In dieser wird die aktuelle Datei vollständig angezeigt (wichtig, wenn mit der Kurzform gearbeitet wird). Wertvoll ist der Status besonders, wenn Dateien markiert werden – zeigt er doch die Gesamtgröße. Ebenfalls damit im Zusammenhang steht der Punkt **Ins move down**. Hier wird festgelegt, ob beim markieren mit <Ins> bzw. <Einfg> der Balkencursor automatisch eine Zeile nach unten soll. Wenn Sie die Uhrzeit gern stets im Blick haben, so kann diese mit **cLock** in der rechten oberen Ecke eingeschaltet werden. Das groß geschriebene L bedeutet, daß auch die Großbuchstaben in der jeweiligen Zeile zur Auswahl herangezogen werden können. Für alle, die statt des integrierten Editors lieber ein eigenes Textverarbeitungsprogramm auf <F4> legen wollen, gibt es den Menüpunkt **Editor**. Er erlaubt die Einbindung eines Textprogramms eigener Wahl, das dann mit <F4> erreichbar ist. Der eingebaute Editor bleibt aber weiterhin über <Alt>-<F4> erhalten. Wenden wir uns als nächstes dem Punkt **Commands** im Setup-Menü zu (siehe Bild 5). Die ersten fünf Eintragungen sind bis auf <Alt>-<F9> zum Umschalten auf den 43-Zeilen-Modus einschließlich ihrer Shortcuts bereits bekannt. Mit **Compare directories** können die Einträge zweier Verzeichnisse verglichen wer-

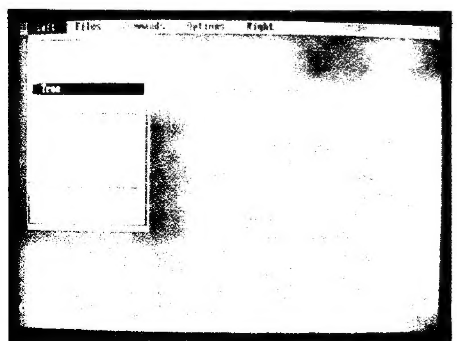


Bild 2 Einstellen des linken Panels im Setup-Menü

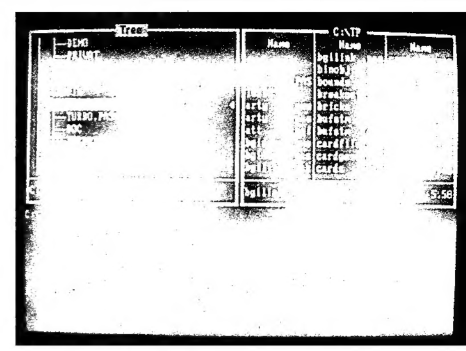


Bild 3 Anzeige im Brief-Format mit Dateibaum

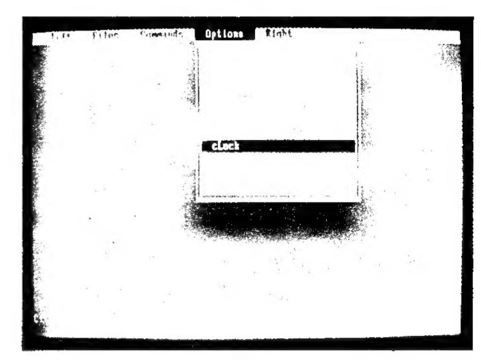


Bild 4 Das Option-Menü

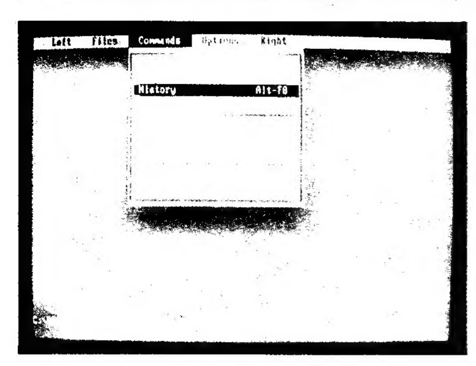


Bild 5 Im Kommando-Menü sind die Shortcuts noch einmal zusammengefaßt

den, etwa um festzustellen, ob beim Kopieren nichts vergessen wurde. Richtig interessant wird es aber bei den letzten beiden Menüpunkten. Sie erlauben das Anlegen und Editieren von Textfiles mit nutzereigenen Aktionen.

Für sich stets wiederholende Eingaben wird dem Nutzer die Möglichkeit gegeben, eigene Menüs zu definieren. Wenn Sie einmal <F2> bedienen, werden Sie feststellen, ob bereits derartige Vereinbarungen in einer Datei Nc.mnu getroffen worden sind und die Auswahl von vordefinierten Aktionen gestatten. Wenn nicht, können Sie sich diese selbst anlegen, wofür der Menüpunkt **Menu file edit** steht. Der Aufbau sieht folgendermaßen aus: Zuerst steht die jeweilige Menüausschrift (muß in der ersten Spalte beginnen), und auf den nächsten Zeilen (etwas eingerückt) folgen die dazugehörigen Kommandos. Hier als Beispiel einmal zwei Menüpunkte:

```
Testprogramm übersetzen
  cd\masm
  masm test;
  link test;
  exe2bin test test.com
  del test.obj
```

Das Beispiel zeigt, daß es sich auch für eine einmalige Testphase lohnen kann, einen Menüpunkt einzurichten. Beim Aufruf von Programmen sollte von der Möglichkeit Gebrauch gemacht werden, einen neuen Pfad einzustellen und Environment-Variablen zu setzen. Vor der Menüausschrift kann auch noch ein Buchstabe oder die Bezeichnung einer Funktionstaste (F1 ... F10) festgelegt werden und dann zur Kurzanwahl dienen. Tragen sie beispielsweise

T: Testprogramm übersetzen

ein, so ist dieser Punkt im **Usermenü** auch kurz mit **t** aktivierbar.

Ähnlich einfache Möglichkeiten der Ausführung von Nutzerkommandos bietet das sogenannte Extension-File (Datei Nc.ext). Hier wird festgelegt, was mit Dateien eines bestimmten Typs getan werden soll, wenn der Nutzer den Balkenkursor auf sie positioniert und <Enter> drückt. Für Exe-, Com- und Bat-Files sind bereits Aktionen festgelegt: Sie werden gestartet. Alle anderen Typen stehen für eigene Festlegungen zur Verfügung.

Die Eintragungen besitzen folgende Syntax:

Erweiterung : Kommando

Als Platzhalter für die aktuell ausgewählte Datei steht dabei ein Ausrufezeichen. Es wird ähnlich benutzt wie die bekannten Joker. So wird für **!** die aktuelle Datei ohne Erweiterung eingesetzt, für **!.!** die Datei mit Erweiterung. Für das aktuelle Laufwerk steht **!:** und für den aktuellen Pfad **!**.

Um also Pascal-Dateien mit dem Typ Pas durch den Kommandozeilencompiler übersetzen zu lassen, ist

pas: tpc !.!

einzutragen. Soll statt dessen die integrierte Entwicklungsumgebung aufgerufen werden und die betreffende Datei geladen werden, so erfordert das beispielsweise den Eintrag:

pas: c:\tp\turbo !:!\!.!

Sowohl die Datei Nc.mnu als auch Nc.ext kann auch mit jedem anderen Texteditor erzeugt werden. Zu erwähnen bleibt noch, daß der Norton Commander nach einer Menüdatei zuerst stets im aktuellen Verzeichnis sucht, so daß jedem Nutzer seine ganz persönliche Einstellung zur Verfügung steht.

Norton und die Maus

Der Norton Commander kann – wie man das von einem modernen Programm erwartet – auch mit der Maus bedient werden. Ob man die Mausbedienung bevorzugt, ist Gewohnheits- und Geschmackssache. Voraussetzung für die Nutzung des kleinen Nagers ist das Laden des Microsoft-Maustreibers. Dazu ist die Datei Mouse.com zu starten. Der Mauskursor ist anschließend als Rechteck (auf einem Farbbildschirm andersfarbig) zu erkennen. Positioniert man ihn auf einem Eintrag des Verzeichnisfensters, so kann dieser durch Drücken der linken Maustaste ausgewählt werden. Das entspricht der Bewegung des Balkenkursors mit den Pfeiltasten. (Anmerkung: Im folgenden wird stets auf die originale Zweitastenmaus von Microsoft Bezug genommen, daneben gibt es auch Exoten mit einer oder drei Tasten.) Die rechte Maustaste wirkt wie <Ins> bzw. <Einfg> und <Del> bzw. <Entf> die Dateien, auf denen der Kursor gerade steht, werden markiert, bzw. die Markierung wird aufgehoben. Sichtbar wird das wie beim Markieren über die Tastatur. Was die Enter-Taste auf der Tastatur, ist der Doppelklick (zweimaliges Drükken kurz hintereinander) der linken Taste für die Maus. Er dient zum Starten von Programmen (bzw. zum Ausführen einer Aktion entsprechend der Programmerweiterung). Mit der Maus kann auch sehr schnell gescrollt werden. Dazu ist der Mauskursor auf die oberste Zeile (die die Überschrift enthält) oder auf den unteren Rand zu setzen und die linke Taste zu drücken. Die Dateieinträge rollen automatisch nach oben bzw. unten. Wird statt dessen die rechte Taste festgehalten, so erfolgt gleichzeitig ein Markieren aller Dateien. Das geht sogar schneller als mit der Tastatur. Fährt man mit dem Kursor in die oberste Zeile, so erscheint automatisch das Setup-Menü (wie bei <F9>), und man kann per linkem Klick die einzelnen Optionen einstellen. Des weiteren ist ein Aktivieren der Menüpunkte am unteren Bildschirmrand (KeyBar genannt) durch einfaches Anklicken möglich.

Die Version 3.0 des Norton Commanders wartet mit einer Reihe weiterer Verbesserungen auf. Zu nennen sind die neuen Komponenten **Commander Link** (zur Verbindung von Rechnern über ein spezielles Kabel, z. B. zum „Füttern" eines Laptops), **Commander Mail** (zum Verschicken von Nachrichten im Netz) sowie **Quick View** zur direkten Unterstützung einer Vielzahl von internen Dateiformaten der gängigen Datenbank-, Kalkulations-, Textverarbeitungs- und Grafikprogramme.

Der Punkt **Configuration** im Option-Menü faßt grundsätzliche Einstellungen zusammen. Positiv anzumerken ist die Hilfe zu jedem Punkt mit <F1>, auffällig auch das schonende Verwandeln des Bildschirms nach einer bestimmten Zeit in einen dunklen Nachthimmel mit ständig aufgehenden und wieder verlöschenden Sternen.

Tafel 1 Tastenkombinationen des Norton Commanders

Taste	Funktion	
<ESC>	Löschen der Kommandozeile, Abbruch einer Aktion	*)
<Tab>	Wechseln des Verzeichnisfensters	
<+>	Markieren von Dateigruppen	
<–>	Aufheben von Gruppenmarkierungen	
<PgUp>/<Bild↑>	Eine Seite nach oben blättern	
<PgDn>/<Bild↓>	Eine Seite nach unten blättern	
<Home>/<Pos1>	Balkenkursor auf ersten Eintrag	
<End>,<Ende>	Balkenkursor auf letzten Eintrag	
<Ins>/<Einfg>	Markieren einer Datei bzw. Aufheben der Markierung	
<F1>	Hilfe	
<F2>	Aufruf UserMenü	
<F3>	Ansehen (View) von Dateien	
<F4>	Editieren von Dateien	
<F5>	Kopieren von Dateien	
<F6>	Umbenennen oder Verschieben	
<F7>	Anlegen von Verzeichnissen	
<F8>	Löschen von Dateien oder Verzeichnissen	
<F9>	Einstellen der Konfiguration (Setup)	
<F10>	Verlassen des Norton Commanders	
<Shift>-<F3>	Ansehen von Dateien mit Abfrage	
<Shift>-<F4>	Anlegen einer neuen Datei	
<Shift>-<F5>	Kopieren von Dateien mit Abfrage	
<Shift>-<F6>	Umbenennen mit Abfrage	
<Shift>-<F8>	Löschen von Dateien mit Abfrage	
<Shift>-<F9>	Speichern der Konfiguration	
<Alt>-<F1>	Laufwerkswechsel im linken Fenster	
<Alt>-<F2>	Laufwerkswechsel im rechten Fenster	
<Alt>-<F3>	Ansehen von dBase-Dateien ohne Dbview.exe	
<Alt>-<F4>	Editieren mit Build-in-Editor	
<Alt>-<F7>	Dateien suchen	
<Alt>-<F8>	Auswahl von Befehlen aus dem History-Stapel	
<Alt>-<F9>	Umschalten auf EGA-Modus (43 Zeilen)	
<Alt>-<F10>	Anzeige des Dateibaums	
<Ctrl>-<Break> /<Strg> <Untbr>	Abbrechen von Gruppenfunktionen	
<Ctrl>-<F1>	Aus-/Einblenden des linken Fensters	
<Ctrl>-<F2>	Aus-/Einblenden des rechten Fensters	
<Ctrl>-<Enter>	Kopieren des Dateinamens in die Eingabezeile	
<Ctrl>-<PgUp>	Wechsel ins übergeordnete Verzeichnis	
<Ctrl>-<\>	Wechsel ins Hauptverzeichnis	
<Ctrl>-<A>	Kursor ein Wort nach links	*)
<Ctrl>-<B>	Hilfstexte (KeyBar) aus- bzw. einblenden	
<Ctrl>-<E>	Rückwärtsblättern im Eingabepuffer	*)
<Ctrl>-<F>	Kursor ein Wort nach rechts	*)
<Ctrl>-<G>	Zeichen an Kursorposition löschen	*)
<Ctrl>-<H>	Zeichen löschen links	*)
<Ctrl>-<I>	wie <Tab>	
<Ctrl>-<J>	wie <Ctrl>-<Enter>	
<Ctrl>-<L>	Statusfenster ein- bzw. ausschalten	
<Ctrl>-<O>	Aus- bzw. Einblenden beider Verzeichnisfenster	
<Ctrl>-<P>	Aus- bzw. Einblenden des nicht aktiven Verzeichnisfensters	
<Ctrl>-<T>	Wort rechts löschen	*)
<Ctrl>-<U>	Vertauschen der Verzeichnisfenster	
<Ctrl>-<W>	Wort löschen links	*)
<Ctrl>-<X>	Vorwärtsblättern im Eingabepuffer	*)
<Ctrl>-<Y>	Löschen der Kommandozeile	*)

*)Aktionen beziehen sich auf die Kommandozeile